

Degree and Sensitivity: tails of two distributions*

Parikshit Gopalan
Microsoft Research
parik@microsoft.com

Rocco A. Servedio[†]
Columbia University
rocco@cs.columbia.edu

Avishay Tal[‡]
IAS, Princeton
avishay.tal@gmail.com

Avi Wigderson[§]
IAS, Princeton
avi@ias.edu

April 27, 2016

Abstract

The *sensitivity* of a Boolean function f is the maximum, over all inputs x , of the number of sensitive coordinates of x (namely the number of Hamming neighbors of x with different f -value). The well-known *sensitivity conjecture* of Nisan (see also Nisan and Szegedy) states that every sensitivity- s Boolean function can be computed by a polynomial over the reals of degree $\text{poly}(s)$. The best known upper bounds on degree, however, are exponential rather than polynomial in s .

Our main result is an approximate version of the conjecture: every Boolean function with sensitivity s can be ϵ -approximated (in ℓ_2) by a polynomial whose degree is $O(s \cdot \log(1/\epsilon))$. This is the first improvement on the folklore bound of s/ϵ . Further, we show that improving the bound to $O(s^c \cdot \log(1/\epsilon)^\gamma)$ for any $\gamma < 1$ and any $c > 0$ will imply the sensitivity conjecture. Thus our result is essentially the best one can hope for without proving the conjecture. We apply our approximation result to obtain a new learning algorithm for insensitive functions, as well as new bounds on the Fourier coefficients and the entropy-influence conjecture for them.

We prove our result via a new “switching lemma for low-sensitivity functions” which establishes that a random restriction of a low-sensitivity function is very likely to have low decision tree depth. This is analogous to the well-known switching lemma for AC^0 circuits. Our proof analyzes the combinatorial structure of the graph G_f of sensitive edges of a Boolean function f . We introduce new parameters of this graph such as *tree sensitivity*, study their relationship, and use them to show that the graph of a function of full degree must be sufficiently complex, and that random restrictions of low-sensitivity functions are unlikely to yield complex graphs.

We postulate a robust analogue of the sensitivity conjecture: if **most** inputs to a Boolean function f have low sensitivity, then **most** of the Fourier mass of f is concentrated on small subsets. We prove a lower bound on tree sensitivity in terms of decision tree depth, and show that a polynomial strengthening of this lower bound implies the robust conjecture. We feel that studying the graph G_f is interesting in its own right, and we hope that some of the notions and techniques we introduce in this work will be of use in its further study.

*The conference version of this paper will appear in CCC’2016 [GSW16a].

[†]Supported by NSF grants CCF-1319788 and CCF-1420349.

[‡]Supported by the Simons Foundation and by NSF grant CCF-1412958.

[§]This research was partially supported by NSF grant CCF-1412958.

1 Introduction

The smoothness of a continuous function captures how gradually it changes locally (according to the metric of the underlying space). For Boolean functions on $\{0, 1\}^n$, a natural analog is *sensitivity*, capturing how many neighbors of a point have different function values. More formally, the *sensitivity* of $f : \{0, 1\}^n \rightarrow \{\pm 1\}$ at input $x \in \{0, 1\}^n$, written $s(f, x)$, is the number of neighbors y of x in the Hamming cube $\{0, 1\}^n$ such that $f(y) \neq f(x)$. The *max sensitivity* of f , written $s(f)$ and often referred to simply as the “sensitivity of f ”, is defined as $s(f) = \max_{x \in \{0, 1\}^n} s(f, x)$. Hence we have $0 \leq s(f) \leq n$ for every $f : \{0, 1\}^n \rightarrow \{\pm 1\}$; while not crucial, it may be helpful to consider this parameter as “low” when e.g. either $s(f) \leq (\log n)^{O(1)}$ or $s(f) \leq n^{o(1)}$ (note that both these notions of “low” are robust up to polynomial factors).

A well known conjecture, sometimes referred to as the “sensitivity conjecture,” states that every smooth Boolean function is computed by a low degree real polynomial, specifically of degree polynomial in its sensitivity. This conjecture was first posed in the form of a question by Nisan [Nis91] and Nisan and Szegedy [NS94] but is now (we feel) widely believed to be true:

Conjecture 1.1. [Nis91, NS94] *There exists a constant c such that every Boolean function f is computed by a polynomial of degree $\deg(f) \leq s(f)^c$.*

Despite significant effort ([KK04, ABG⁺14, AP14, APV15, AV15]) the best upper bound on degree in terms of sensitivity is exponential. Recently several consequences of Conjecture 1.1, e.g. that every f has a *formula* of depth at most $\text{poly}(s(f))$, have been unconditionally established in [GNS⁺16]. Nisan and Szegedy proved the converse, that every Boolean function satisfies $s(f) = O(\deg(f)^2)$.

In this work, we make progress on Conjecture 1.1 by showing that functions with low max sensitivity are very well *approximated* (in ℓ_2) by low-degree polynomials. We exponentially improve the folklore $O(s/\epsilon)$ degree bound (which follows from average sensitivity and Markov’s inequality) by replacing the $1/\epsilon$ error dependence with $\log(1/\epsilon)$. The following is our main result:

Theorem 1.2. *For any Boolean function $f : \{0, 1\}^n \rightarrow \{\pm 1\}$ and any $\epsilon > 0$, there exists a polynomial $p : \{0, 1\}^n \rightarrow \mathbb{R}$ with $\deg(p) \leq O(s(f) \cdot \log(1/\epsilon))$ such that $\mathbb{E}_{x \in \{0, 1\}^n} [|p(x) - f(x)|^2] \leq \epsilon$.¹*

One might wonder if the dependence on ϵ can be improved further. We observe that a bound of $O(s^c \cdot \log(1/\epsilon)^\gamma)$ for any constants $\gamma < 1$ and $c > 0$ implies Conjecture 1.1. Thus Theorem 1.2 gets essentially the best bound one can hope for without proving Conjecture 1.1. Furthermore, we show that a bound of $O(s \cdot \log(1/\epsilon)^\gamma)$ for a constant $\gamma < 1$ does not hold for a family of Boolean functions based on the Hamming code. Hence, for $c = 1$, Theorem 1.2 is essentially tight.

En route to proving this result, we make two related contributions which we believe are interesting in themselves:

- Formulating a robust variant of the sensitivity conjecture (proving which would generalize Theorem 1.2).
- Defining and analyzing some natural graph-theoretic complexity measures, essential to our proof and which we believe may hold the key to progress on the original and robust sensitivity conjectures.

¹The conference version of this paper [GSW16a] proves a weaker bound of $O(s(f)(\log(1/\epsilon)^3))$.

1.1 A robust variant of the sensitivity conjecture

A remarkable series of developments, starting with [Nis91], showed that real polynomial degree is an extremely versatile complexity measure: it is polynomially related to many other complexity measures for Boolean functions, including PRAM complexity, block sensitivity, certificate complexity, deterministic/randomized/quantum decision tree depth, and approximating polynomial degree (see [BdW02, HKP11] for details on many of these relationships). Arguably the one natural complexity measure that has defied inclusion in this equivalence class is sensitivity. Thus, there are many equivalent formulations of Conjecture 1.1; indeed, Nisan’s original formulation was in terms of sensitivity versus block sensitivity [Nis91].

Even though progress on it has been slow, over the years Conjecture 1.1 has become a well-known open question in the study of Boolean functions. It is natural to ask *why* this is an important question: will a better understanding of sensitivity lead to new insights into Boolean functions that have eluded us so far? Is sensitivity qualitatively different from the other concrete complexity measures that we already understand?

We believe that the answer is yes, and in this paper we make the case that Conjecture 1.1 is just the (extremal) tip of the iceberg: it hints at deep connections between the *combinatorial* structure of a Boolean function f , as captured by the graph G_f of its sensitive edges in the hypercube, and the *analytic* structure, as captured by its Fourier expansion. This connection is already the subject of some of the key results in the analysis of Boolean functions, such as [KKL88, Fri98], as well as important open problems like the “entropy-influence” conjecture [FK96] and its many consequences.

Given any Boolean function f , we conjecture a connection between the distribution of the sensitivity of a random vertex in $\{0, 1\}^n$ and the distribution of f ’s Fourier mass. This conjecture, which is an important motivation for the study in this paper, is stated informally below:

Robust Sensitivity Conjecture (Informal Statement): *If **most** inputs to a Boolean function f have low sensitivity, then **most** of the Fourier mass of f is concentrated on small subsets.*

Replacing both occurrences of **most** by *all* we recover Conjecture 1.1, and hence the statement may be viewed as a robust formulation of the sensitivity conjecture. Theorem 1.2 corresponds to replacing the first **most** by *all*. There are natural classes of functions which do not have low max sensitivity, but for which most vertices have low sensitivity; the robust sensitivity conjecture is relevant to these functions while the original sensitivity conjecture is not. (A prominent example of such a class is AC^0 , for which the results of [LMN93] establish a weak version of the assumption (that most inputs have low sensitivity) and the results of [LMN93, Tal14b] establish a strong version of the conclusion (Fourier concentration).)

In order to formulate a precise statement, for a given Boolean function $f : \{0, 1\}^n \rightarrow \{\pm 1\}$ we consider the random experiment which samples from the following two distributions:

1. The Sensitivity distribution: sample a uniform random vertex $\mathbf{x} \in \{0, 1\}^n$ and let $\mathbf{s} = s(f, \mathbf{x})$.
2. The Fourier distribution: sample a subset $\mathbf{T} \subset [n]$ with probability $\hat{f}(\mathbf{T})^2$ and let $\mathbf{d} = |\mathbf{T}|$.

We conjecture a close relation between the k^{th} moments of these random variables:

Conjecture 1.3 (Robust Sensitivity Conjecture). *For all Boolean functions f and all integers $k \geq 1$, there is a constant a_k such that $\mathbb{E}[\mathbf{d}^k] \leq a_k \mathbb{E}[\mathbf{s}^k]$.*

The key here is that there is no dependence on n . To see the connection with the informal statement above, if a function has low sensitivity for most $x \in \{0, 1\}^n$, then it must have bounded k^{th} sensitivity moments for fairly large k ; in such a case, Conjecture 1.3 implies a strong Fourier concentration bound by Markov’s inequality. The classical Fourier expansion for average sensitivity tells us that when $k = 1$, $\mathbb{E}[s] = \mathbb{E}[d]$. It is also known that $\mathbb{E}[s^2] = \mathbb{E}[d^2]$ (see e.g. [CKLS15, Lemma 3.5]), but equality does not hold for $k \geq 3$. Conjecture 1.3 states that if we allow constant factors depending on k , then one direction still holds.

It is clear that Conjecture 1.3 (with a_k a not-too-rapidly-growing function of k) is a strengthening of our Theorem 1.2. To see its relation to Conjecture 1.1 observe that Conjecture 1.1 implies that for $k \rightarrow \infty$, $\mathbb{E}[d^k] \leq a^k (\mathbb{E}[s^k])^b$ for constants a, b . On the other hand, via Markov’s inequality, Conjecture 1.3 only guarantees Fourier concentration rather than small degree for functions with small sensitivity. Thus the robust version Conjecture 1.3 seems incomparable to Conjecture 1.1.

It is possible that the reverse direction of the robust conjecture also holds: for every k there exists a'_k such that $\mathbb{E}[s^k] \leq a'_k \mathbb{E}[d^k]$; settling this is an intriguing open question.

Both our proof of Theorem 1.2, and our attempts at Conjecture 1.3, follow the same general path. We apply random restrictions, which reduces these statements to analyzing some natural new graph-theoretic complexity measures of Boolean functions. These measures are relaxations of sensitivity: they look for occurrences of various subgraphs in the sensitivity graph, rather than just high degree vertices. We establish (and conjecture) connections between different graph-theoretic measures and decision tree depth (see Theorem 5.4, which relates decision tree depth and the length of “proper walks”, and Conjecture 4.10, which conjectures a relation between “tree sensitivity” and decision tree depth). These connections respectively enable the proof of Theorem 1.2 and provide a simple sufficient condition implying Conjecture 1.3, which suffices to prove the conjecture for $k = 3$ and 4. We elaborate on this in the next subsection. We believe that these new complexity measures are interesting and important in their own right, and that understanding them better may lead to progress on Conjecture 1.1.

1.2 Random restrictions and graph-theoretic complexity measures

In this subsection we give a high level description of our new complexity measures and perspectives on the sensitivity graph and of how we use them to approach Conjecture 1.3 and prove Theorem 1.2. As both have the same conclusion, namely strong Fourier concentration, we describe both approaches together until they diverge. This leads to analyzing two different graph parameters (as we shall see, the stronger assumption of Theorem 1.2 allows the use of a weaker graph parameter that we can better control).

First we give a precise definition of the *sensitivity graph*: to every Boolean function f we associate a graph G_f whose vertex set is $\{0, 1\}^n$ and whose edge set E consists of all edges (x, y) of the hypercube that have $f(x) \neq f(y)$. Each edge is labelled by the coordinate in $[n]$ at which x and y differ. The degree of vertex x is exactly $s(f, x)$, and the maximum degree of G_f is $s(f)$.

The starting point of our approach is to reinterpret the moments of the degree and sensitivity distributions of f in terms of its random restrictions. Let $\mathcal{R}_{k,n}$ denote the distribution over random restrictions that leave exactly k of the n variables unset and set the rest uniformly at random. We first show, in Section 3, that the k^{th} moment of the sensitivity distribution controls the probability that a random restriction f_ρ of f , where $\rho \leftarrow \mathcal{R}_{k,n}$, has full sensitivity (Theorem 3.1). Similarly, moments of the Fourier distribution capture the event that f_ρ has full degree (Theorem 3.2). (We note that Tal has proved a result of a similar flavor; [Tal14a, Theorem 3.2] states that strong Fourier

concentration of f implies that random restrictions of f are unlikely to have high degree.)

Random restrictions under sensitivity moment bounds: Towards Conjecture 1.3

Given Theorems 3.1 and 3.2, Conjecture 1.3 may be rephrased as saying that if a function f has low sensitivity moments, then a random restriction f_ρ is unlikely to have full degree. Some intuition supporting this statement is that the sensitivity graphs of functions with full degree should be “complex” (under some suitable complexity measure), whereas the graph of f_ρ is unlikely to be “complex” if f has low sensitivity moments. More precisely, the fact that G_f has no (or few) vertices of high degree suggests that structures with many sensitive edges in distinct directions will not survive a random restriction.

Some evidence supporting this intuition is given by Theorem 3.1, which tells us that if f has low sensitivity moments then f_ρ is unlikely to have full sensitivity. If full degree implied full sensitivity then we would be done, but this is false as witnessed e.g. by the three-variable majority function and by composed variants of it. (Conjecture 1.1 asserts that the gap between degree and sensitivity is at most polynomial, but of course we do not want to invoke the conjecture!) This leads us in Section 4 to consider our first relaxation of sensitivity, which we call *tree-sensitivity*. To motivate this notion, note that a vertex with sensitivity k is simply a star with k edges in the sensitivity graph. We relax the star requirement and consider all *sensitive trees*: trees of sensitive edges (i.e. edges in G_f) where every edge belongs to a *distinct* coordinate direction (as is the case, of course, for a star). Analogous to the usual notion of sensitivity, the tree sensitivity of f at x is the size of the largest sensitive tree containing x , and the tree sensitivity of f is the maximum tree sensitivity of f at any vertex.

Theorem 4.11 shows that the sensitivity moments of f control the probability that f_ρ has full tree sensitivity. Its proof crucially uses a result by Sidorenko [Sid94] on counting homomorphisms to trees. Theorem 4.11 would immediately imply Conjecture 1.3 if every function of degree k must have tree sensitivity k . (This is easily verified for $k = 3, 4$, which, as alluded to in the previous subsection, gives Conjecture 1.3 for those values of k .) The best we can prove, though, is a tree sensitivity lower bound of $\Omega(\sqrt{k})$ (Theorem 4.9); the proof of this lower bound uses notions of maximality and “shifting” of sensitive trees that we believe may find further application in the study of tree sensitivity. We conjecture that full degree does imply full tree sensitivity, implying Conjecture 1.3. This is a rare example where having a precise bound between the two complexity measures (rather than a polynomial relationship) seems to be important.

Random restrictions under max sensitivity bounds: Proving Theorem 1.2

Next, we aim to prove *unconditional* moment bounds on the Fourier distribution of low sensitivity functions and thereby obtain Theorem 1.2. Towards this goal, in Section 5 we relax the notion of tree sensitivity and study certain walks in the Boolean hypercube that we call *proper walks*: these are walks such that every time a coordinate direction is explored for the first time, it is along a sensitive edge. We show in Theorem 5.4 that having full decision tree depth implies the existence of a very short (length $O(n)$) proper walk containing sensitive edges along every coordinate. In Lemma 5.5, we analyze random restrictions to show that such a structure is unlikely to survive in the remaining subcube of unrestricted variables. This may be viewed as a “switching lemma for low-sensitivity functions”, which again may be independently interesting (note that strictly speaking this result is not about switching from a DNF to a CNF or vice versa, but rather it upper

bounds the probability that a restricted function has large decision tree depth, in the spirit of standard “switching lemmas”). It yields Theorem 1.2 via a rather straightforward argument. The analysis requires an upper bound on the maximum sensitivity because we do not know an analogue of Sidorenko’s theorem for proper walks.

1.3 Some high-level perspective

An important goal of this work is to motivate a better understanding of the combinatorial structure of the sensitivity graph G_f associated with a Boolean function. In our proofs other notions suggest themselves beyond tree sensitivity and proper walks, most notably the *component dimension* of the graph, which may be viewed as a further relaxation of sensitivity. Better relating these measures to decision tree depths, as well as to each other, remains intriguing, and in our view promising, for making progress on Conjecture 1.1 and Conjecture 1.3 and for better understanding Boolean functions in general. We hope that some of the notions and techniques we introduce in this work will be of use to this goal.

Another high level perspective relates to “switching lemmas”. As mentioned above, we prove here a new result of this kind, showing that under random restrictions low sensitivity functions have low decision tree depth with high probability. The classical switching lemma shows the same for small width DNF (or CNF) formulas (and hence for AC^0 circuits as well). Our proof is quite different than the standard proofs, as it is essentially based on the combinatorial parameters of the sensitivity graph. Let us relate the assumptions of both switching lemmas. On the one hand, by the sensitivity Conjecture 1.1 (which we can’t use, and want to prove), low sensitivity should imply low degree and hence low decision tree depth and small DNF width. On the other hand, small DNF width (indeed small, shallow circuits) imply (by [LMN93]) low *average* sensitivity, which is roughly the assumption of the robust sensitivity Conjecture 1.3. As it turns out, we can use our combinatorial proof of our switching lemma to derive a somewhat weaker form of the original switching lemma, and also show that the same combinatorial assumption (relating tree sensitivity to decision tree depth) which implies Conjecture 1.3 would yield a nearly tight form of the original switching lemma. This lends further motivation to the study of these graph parameters.

Another conjecture formalizing the maxim that *low sensitivity implies Fourier concentration* is the celebrated Entropy-Influence conjecture of Freidgut and Kalai [FK96] which posits the existence of a constant C such that $\mathbb{H}(\mathbf{T}) \leq C \mathbb{E}[s]$ where $\mathbb{H}(\cdot)$ denotes the entropy function of a random variable.² The conjecture states that functions with low sensitivity on average (measured by $\mathbb{E}[s] = \mathbb{E}[d]$) have their Fourier spectrum concentrated on a few coefficients, so that the entropy of the Fourier distribution is low. However, unlike in Conjecture 1.3 the degree of those coefficients does not enter the picture.

Organization. We present some standard preliminaries and notation in Section 2. Section 3 proves Theorems 3.1 and 3.2 which show that degree and sensitivity moments govern the degree and sensitivity respectively of random restrictions.

²Recall that the entropy $\mathbb{H}(\mathbf{T})$ of the random variable \mathbf{T} is

$$\mathbb{H}(\mathbf{T}) = \sum_{T \subseteq [n]} \Pr[\mathbf{T} = T] \log_2 \frac{1}{\Pr[\mathbf{T} = T]}.$$

In Section 4 we study tree sensitivity. Section 4.1 relates it to other complexity measures, while Section 4.2 relates tree sensitivity of a random restriction to the sensitivity moments. Section 5 introduces proper walks and uses them to show Fourier concentration for low sensitivity functions. We define proper walks in Section 5.1, and use them to analyze random restrictions of low-sensitivity functions in section 5.2. We prove Theorem 1.2 in 5.3 and analyze its tightness in 5.4. Section 5 uses results from Section 4.1 but is independent of the rest of Section 4.

We derive applications to learning in Section 6.1, to the Entropy-Influence conjecture in Section 6.2, and present an approach to proving the Switching lemma for DNFs via sensitivity moments in Section 6.3. In Section 7 we present a family of functions that demonstrates the tightness of Theorem 1.2, and some additional examples, and highlight some open problems in Section 8.

2 Preliminaries

The Fourier distribution. Let $f : \{0,1\}^n \rightarrow \{\pm 1\}$ be a Boolean function. We define the usual inner product on the space of such functions by $\langle f, g \rangle = \mathbb{E}_{\mathbf{x} \leftarrow \{0,1\}^n} [f(\mathbf{x})g(\mathbf{x})]$. For $S \subseteq [n]$ the parity function χ_S is $\chi_S(x) = (-1)^{\sum_{i \in S} x_i}$. The Fourier expansion of f is given by $f(x) = \sum_{S \subseteq [n]} \hat{f}(S) \chi_S(x)$, where $\hat{f}(S) = \langle f, \chi_S \rangle$. By Parseval's identity we have $\sum_{S \subseteq [n]} \hat{f}(S)^2 = 1$. This allows us to view any Boolean function f as inducing a probability distribution \mathcal{D}_f on subsets $S \subseteq [n]$, given by $\Pr_{\mathbf{R} \leftarrow \mathcal{D}_f} [\mathbf{R} = S] = \hat{f}(S)^2$. We refer to this as the *Fourier distribution*. We define $\text{supp}(f) \subseteq 2^{[n]}$ as $\text{supp}(f) = \{S \subseteq [n] : \hat{f}(S)^2 \neq 0\}$. The Fourier expansion of f can be viewed as expressing S as a multilinear polynomial in x_1, \dots, x_n , so that $\deg(f) = \max_{S \in \text{supp}(f)} |S|$. Viewing \mathcal{D}_f as a probability distribution on $2^{[n]}$, we define the following quantities which we refer to as “influence moments” of f :

$$\mathbb{I}^k[f] = \mathbb{E}_{\mathbf{R} \leftarrow \mathcal{D}_f} [|\mathbf{R}|^k] = \sum_S \hat{f}(S)^2 |S|^k, \quad (1)$$

$$\mathbb{I}^{\leq k}[f] = \mathbb{E}_{\mathbf{R} \leftarrow \mathcal{D}_f} \left[\prod_{i=0}^{k-1} (|\mathbf{R}| - i) \right] = \sum_{|S| \geq k} \hat{f}(S)^2 \prod_{i=0}^{k-1} (|S| - i). \quad (2)$$

We write $\deg_\epsilon(f)$ to denote the minimum k such that $\sum_{S \subseteq [n]; |S| \geq k} \hat{f}(S)^2 \leq \epsilon$. It is well known that $\deg_\epsilon(f) \leq k$ implies the existence of a degree k polynomial g such that $\mathbb{E}_{\mathbf{x}} [(f(\mathbf{x}) - g(\mathbf{x}))^2] \leq \epsilon$; g is obtained by truncating the Fourier expansion of f to level k .

The sensitivity distribution. We use $d(\cdot, \cdot)$ to denote Hamming distance on $\{0,1\}^n$. The n -dimensional hypercube H_n is the graph with vertex set $V = \{0,1\}^n$ and $\{x, y\} \in E$ if $d(x, y) = 1$. For $x \in \{0,1\}^n$, let $N(x)$ denote its neighborhood in H_n . As described in Section 1, the *sensitivity* of a function f at point x is defined as $s(f, x) = |\{y \in N(x) : f(x) \neq f(y)\}|$, and the (worst-case) sensitivity of f , denoted $s(f)$, is defined as $s(f) = \max_{x \in \{0,1\}^n} s(f, x)$. Analogous to (1) and (2), we define the quantities $s^k(f)$ and $s^{\leq k}(f)$ which we refer to as “sensitivity moments” of f :

$$s^k(f) = \mathbb{E}_{\mathbf{x} \leftarrow \{0,1\}^n} [s(f, \mathbf{x})^k], \quad s^{\leq k}(f) = \mathbb{E}_{\mathbf{x} \leftarrow \{0,1\}^n} \left[\prod_{i=0}^{k-1} (s(f, \mathbf{x}) - i) \right]. \quad (3)$$

With this notation, we can restate Conjecture 1.3 (with a small modification) as

Conjecture. (Conjecture 1.3 restated) For every k , there exists constants a_k, b_k such that $\mathbb{I}^k(f) \leq a_k s^k(f) + b_k$.

The reason for the additive constant b_k is that for all non-negative integers x , we have

$$\prod_{i=0}^{k-1} (x - i) \leq x^k \leq e^k \prod_{i=0}^{k-1} (x - i) + k^k.$$

Hence allowing the additive factor lets us freely interchange \mathbb{I}^k with $\mathbb{I}^{\underline{k}}$ and s^k with $s^{\underline{k}}$ in the statement of the Conjecture. We note that $\mathbb{I}^1[f] = \mathbb{I}^{\underline{1}}[f] = s^1(f) = s^{\underline{1}}(f)$, and as stated earlier it is not difficult to show that $\mathbb{I}^2[f] = s^2(f)$ (see e.g. Lemma 3.5 of [CKLS15]). However, in general $\mathbb{I}^k(f) \neq s^k(f)$ for $k \geq 3$ (as witnessed, for example, by the AND function).

Some other complexity measures. We define $\dim(f)$ to be the number of variables that f depends on and $\text{dt}(f)$ to be the smallest depth of a deterministic decision tree computing f . In particular $f : \{0, 1\}^n \rightarrow \{\pm 1\}$ has $\dim(f) = n$ iff f is sensitive to every co-ordinate, and has $\text{dt}(f) = n$ iff f is evasive. It is easy to see that $\deg(f) \leq \text{dt}(f) \leq \dim(f)$ and $s(f) \leq \text{dt}(f)$.

3 Random restrictions and moments of degree and sensitivity

We write $\mathcal{R}_{k,n}$ to denote the set of all restrictions that leave exactly k variables live (unset) out of n . A restriction $\rho \in \mathcal{R}_{k,n}$ is viewed as a string in $\{0, 1, \star\}^n$ where $\rho_i = \star$ for exactly the k live variables. We denote the set of live variables by $L(\rho)$, and we use $f_\rho : \{0, 1\}^{L(\rho)} \rightarrow \{\pm 1\}$ to denote the resulting restricted function. We use $C(\rho) \subseteq \{0, 1\}^n$ to denote the subcube consisting of all possible assignments to variables in $L(\rho)$. We sometimes refer to “a random restriction $\rho \leftarrow \mathcal{R}_{k,n}$ ” to indicate that ρ is selected uniformly at random from $\mathcal{R}_{k,n}$.

A random restriction $\rho \leftarrow \mathcal{R}_{k,n}$ can be chosen by first picking a set $\mathbf{K} \subset [n]$ of k co-ordinates to set to \star and then picking $\rho_{\mathbf{K}} \in \{0, 1\}^{[n] \setminus \mathbf{K}}$ uniformly at random. Often we will pick both $\mathbf{x} \in \{0, 1\}^n$ and $\mathbf{K} \subset [n]$ of size k independently and uniformly at random. This is equivalent to sampling a random restriction ρ and a random point \mathbf{y} within the subcube $C(\rho)$.

The following two theorems show that $\mathbb{I}^{\underline{k}}[f]$ captures the degree of f_ρ , whereas $s^{\underline{k}}(f)$ captures its sensitivity.

Theorem 3.1. Let $f : \{0, 1\}^n \rightarrow \{\pm 1\}$, $\rho \leftarrow \mathcal{R}_{k,n}$, and $1 \leq j \leq k$. Then

$$\frac{s^{\underline{j}}(f)}{n^j} \approx \frac{s^j(f)}{\prod_{i=0}^{j-1} (n - i)} \leq \Pr_{\rho \leftarrow \mathcal{R}_{k,n}} [s(f_\rho) \geq j] \leq \frac{2^k s^{\underline{j}}(f) \binom{k}{j}}{\prod_{i=0}^{j-1} (n - i)} \approx \frac{2^k s^{\underline{j}}(f) \binom{k}{j}}{n^j}. \quad (4)$$

Proof. Consider the bipartite graph in which the vertices X on the left are all j -edge stars S in G_f , the vertices Y on the right are all restrictions $\rho \in \mathcal{R}_{k,n}$, and an edge connects S and ρ if the star S lies in the subcube $C(\rho)$ specified by the restriction ρ . The desired probability $\Pr_{\rho \in \mathcal{R}_{k,n}} [s(f_\rho) \geq j]$ is the fraction of nodes in Y that are incident to at least one edge.

The number of nodes on the left is equal to

$$|X| = \sum_{x \in \{0, 1\}^n} \binom{s(f, x)}{j} = \frac{2^n s^{\underline{j}}(f)}{j!}.$$

The degree of each node S on the left is exactly $\binom{n-j}{k-j}$, since if S is adjacent to ρ then j of the k elements of $L(\rho)$ must correspond to the j edge coordinates of S and the other $k-j$ elements of $L(\rho)$ can be any of the $n-j$ remaining coordinates (note that the non- \star coordinates of ρ are completely determined by S). On the right, a restriction $\rho \in \mathcal{R}_{k,n}$ is specified by a set $L(\rho)$ of k live co-ordinates where $\rho_i = \star$, and a value $\rho_i \in \{0, 1\}$ for the other coordinates, so $|Y| = |\mathcal{R}_{k,n}| = \binom{n}{k} 2^{n-k}$. We thus have

$$\Pr_{\rho \leftarrow \mathcal{R}_{k,n}} [s(f_\rho) \geq j] \leq \frac{\text{total \# of edges into } Y}{|Y|} = \frac{\left(\frac{2^n s^j(f)}{j!}\right) \cdot \binom{n-j}{k-j}}{\binom{n}{k} 2^{n-k}} = \frac{2^k s^j(f) \binom{k}{j}}{\prod_{i=0}^{j-1} (n-i)}.$$

For the lower bound, in order for S to lie in $C(\rho)$ the root of S must belong to $C(\rho)$ (2^k choices) and all edges of S must correspond to elements of $L(\rho)$ ($\binom{k}{j}$ choices), so the maximum degree of any $\rho \in Y$ is $2^k \binom{k}{j}$. Hence we have

$$\Pr_{\rho \leftarrow \mathcal{R}_{k,n}} [s(f_\rho) \geq j] \geq \frac{\frac{(\text{total \# of edges into } Y)}{(\text{max degree of any } \rho \in Y)}}{|Y|} = \frac{\left(\frac{2^n s^j(f)}{j!}\right) \cdot \binom{n-j}{k-j}}{2^k \binom{k}{j} \cdot \binom{n}{k} 2^{n-k}} = \frac{s^j(f)}{\prod_{i=0}^{j-1} (n-i)},$$

□

Theorem 3.2. ³ Let $f : \{0, 1\}^n \rightarrow \{\pm 1\}$ and $\rho \leftarrow \mathcal{R}_{k,n}$. Then

$$\frac{\mathbb{I}^k(f)}{n^k} \approx \frac{\mathbb{I}^k(f)}{\prod_{i=0}^{k-1} (n-i)} \leq \Pr_{\rho \leftarrow \mathcal{R}_{k,n}} [\deg(f_\rho) = k] \leq \frac{2^{2k-2} \mathbb{I}^k(f)}{\prod_{i=0}^{k-1} (n-i)} \approx \frac{2^{2k-2} \mathbb{I}^k(f)}{n^k}. \quad (5)$$

Proof. We first fix $K \subseteq [n]$ and consider the restricted function f_ρ that results from a random choice of $\mathbf{y} = \rho_{\bar{K}} \in \{0, 1\}^{[n] \setminus K}$. The degree k Fourier coefficient of f_ρ equals $\hat{f}_\rho(K)$ and is given by

$$\hat{f}_\rho(K) = \sum_{S \subseteq [n] \setminus K} \hat{f}(S \cup K) \chi_S(\mathbf{y}).$$

Hence we have

$$\mathbb{E}_{\mathbf{y}} [\hat{f}_\rho(K)^2] = \sum_{S \subseteq [n] \setminus K} \hat{f}(S \cup K)^2,$$

and hence over a random choice of \mathbf{K} , we have

$$\mathbb{E}_{\rho} [\hat{f}_\rho(\mathbf{K})^2] = \sum_{S \subseteq [n]} \mathbb{E}_{\rho} [\mathbf{1}(\mathbf{K} \subseteq S)] \hat{f}(S)^2 = \sum_{S \subseteq [n]} \frac{\prod_{i=0}^{k-1} (|S| - i)}{\prod_{i=0}^{k-1} (n - i)} \hat{f}(S)^2 = \frac{\mathbb{I}^k[f]}{\prod_{i=0}^{k-1} (n - i)}. \quad (6)$$

Note that $\deg(f_\rho) = k$ iff $\widehat{f}_\rho(\mathbf{K})^2 \neq 0$. Further, when it is non-zero $\widehat{f}_\rho(\mathbf{K})^2$ lies in the range $[2^{-(2k-2)}, 1]$, since a non-zero Fourier coefficient in a k -variable Boolean function has magnitude at least 2^{-k+1} . Hence we have

$$2^{-2k+2} \Pr_{\rho} [\widehat{f}_\rho(\mathbf{K})^2 \neq 0] \leq \mathbb{E}_{\rho} [\widehat{f}_\rho(\mathbf{K})^2] \leq \Pr_{\rho} [\widehat{f}_\rho(\mathbf{K})^2 \neq 0] \quad (7)$$

which gives the desired bound when plugged into Equation (6). □

³The upper bound in the following theorem is essentially equivalent to Theorem 3.2 of [Tal14a], while the lower bound is analogous to [LMN93]. The only difference is in the family of restrictions.

Conjecture 1.3 revisited: An easy adaptation of the Theorem 3.2 argument gives bounds on $\Pr_{\rho \leftarrow \mathcal{R}_{k,n}}[\deg(f_\rho) \geq j]$. Given these bounds, Conjecture 1.3 implies that for any $j \leq k$,

$$\Pr_{\rho \leftarrow \mathcal{R}_{k,n}}[\deg(f_\rho) \geq j] \leq a_k \Pr_{\rho \leftarrow \mathcal{R}_{k,n}}[s(f_\rho) \geq j] + o_n(1).$$

Indeed, by specifying the $o_n(1)$ term, we can get a reformulation of Conjecture 1.3. This formulation has an intuitive interpretation: *gap examples exhibiting low sensitivity but high degree are not robust to random restrictions*. Currently, we do not know how to upper bound $\deg(f)$ by a polynomial in $s(f)$, indeed we do know of functions f where $\deg(f) \geq s(f)^2$. But the Conjecture implies that if we hit any function f with a random restriction, the probability that the restriction has large degree can be bounded by the probability that it has large sensitivity. Thus the conjecture predicts that these gaps do not survive random restrictions in a rather strong sense.

Implications for AC^0 : For functions with small AC^0 circuits, a sequence of celebrated results culminating in the work of Håstad [Hås86] gives upper bounds on $\Pr[\text{dt}(f_\rho) \geq j]$. Since $\Pr[\text{dt}(f_\rho) \geq j] \geq \Pr[\deg(f_\rho) \geq j]$, we can plug these bounds into Theorem 3.2 to get upper bounds on the Fourier moments, and derive a statement analogous to [LMN93, Lemma 7], [Tal14b, Theorem 1.1] on the Fourier concentration of functions in AC^0 .

Similarly $\Pr[\text{dt}(f_\rho) \geq j] \geq \Pr[s(f_\rho) \geq j]$, so via this approach Theorem 3.1 gives upper bounds on the sensitivity moments, and hence sensitivity tail bounds for functions computed by small AC^0 circuits. This can be viewed as an extension of [LMN93, Lemma 12], which bounds the average sensitivity (first moment) of such functions. For depth 2 circuits, such tail bounds are given by the satisfiability coding lemma [PPZ97], but we believe these are the first such bounds for depth 3 and higher. As this is not the focus of our current work, we leave the details to the interested reader.

4 Tree sensitivity

In this section we study the occurrence of trees of various types in the sensitivity graph G_f , by defining a complexity measure called tree sensitivity. We study its relation to other complexity measures like decision tree depth.

Definition 4.1. A set $S \subseteq \{0,1\}^n$ induces a sensitive tree T in G_f if (i) the points in S induce the (non-trivial) tree T in the Boolean hypercube; (ii) every edge induced by S is a sensitive edge for f , i.e. belongs to $E(G_f)$; and (iii) each induced edge belongs to a distinct co-ordinate direction.

Given a fixed function f , a sensitive tree T is completely specified by the set $V(T)$ of its vertices. We can think of each edge $e \in E(T)$ as being labelled by the coordinate $\ell(e) \in [n]$ along which f is sensitive, so every edge has a distinct label. Let $\ell(T)$ denote the set of all edge labels that occur in T . We refer to $|\ell(T)|$ as the *size* of T , and observe that it lies in $\{1, \dots, n\}$. We note that $|V(T)| = |\ell(T)| + 1$ by the tree property. Further, any two vertices in $V(T)$ differ on a subset of coordinates in $\ell(T)$. Hence the set $V(T)$ lies in a subcube spanned by coordinates in $\ell(T)$, and all points in $V(T)$ agree on all the coordinates in $\overline{\ell(T)} \stackrel{\text{def}}{=} [n] \setminus \ell(T)$.

Definition 4.2. For $x \in \{0,1\}^n$, the tree-sensitivity of f at x , denoted $\text{ts}(f, x)$, is the maximum of $|\ell(T)|$ over all sensitive trees T such that $x \in V(T)$. We define the tree-sensitivity of f as $\text{ts}(f) = \max_{x \in \{0,1\}^n} \text{ts}(f, x)$.

Note that a vertex and all its sensitive neighbors induce a sensitive tree (which is a star). Thus one can view tree-sensitivity as a generalization of sensitivity, and hence we have that $\text{ts}(f) \geq \text{s}(f)$. Lemma 7.1 will show that $\text{ts}(f)$ can in fact be exponentially larger than both $\text{s}(f)$ and $\text{dt}(f)$ (the decision tree depth of f), and thus it cannot be upper bounded by some polynomial in standard measures like decision tree depth, degree, or block sensitivity. However, Theorem 4.9, which we prove in the next subsection, gives a polynomial lower bound.

4.1 Tree sensitivity and decision tree depth

A sensitive tree T is *maximal* if there does not exist any sensitive tree T' with $V(T) \subsetneq V(T')$. In this subsection we study maximal sensitive trees using a “shifting” technique, introduce the notion of an “orchard” (a highly symmetric configuration of isomorphic sensitive trees that have been shifted in all possible ways along their insensitive coordinates), and use these notions to prove Theorem 4.9, which lower bounds tree sensitivity by square root of decision tree depth.

The *support* of a vector $v \in \{0, 1\}^n$, denoted $\text{supp}(v)$, is the set $\{i \in [n] : v_i = 1\}$. For $x, v \in \{0, 1\}^n$, $x \oplus v$ denotes the coordinatewise xor. Given a set $S \subseteq \{0, 1\}^n$, let $S \oplus v = \{x \oplus v : x \in S\}$.

Definition 4.3. Let v be a vector supported on $\overline{\ell(T)}$ where T is a sensitive tree in G_f . We say that T can be shifted by v if $f(x) = f(x \oplus v)$ for all $x \in V(T)$.

If T can be shifted by v then $V(T) \oplus v$ also induces a sensitive tree which we denote by $T \oplus v$. Mapping x to $x \oplus v$ gives an isomorphism between T and $T \oplus v$ which preserves both adjacency and edge labels, and in particular we have $\ell(T \oplus v) = \ell(T)$.

We have the following characterization of maximality (both directions follow easily from the definitions of maximality and of shifting by the unit basis vector e_i):

Lemma 4.4. A sensitive tree T is maximal if and only if it can be shifted by e_i for all $i \in \overline{\ell(T)}$ (equivalently, if none of the vertices in $V(T)$ is sensitive to any coordinate in $\ell(T)$).

The notion of maximality allows for a “win-win” analysis of sensitive trees: for each co-ordinate $i \in \overline{\ell(T)}$, we can either increase the size of the tree by adding an edge in direction i , or we can shift by e_i to get an isomorphic copy of the tree. Repeating this naturally leads to the following definition.

Definition 4.5. Let T be a sensitive tree that can be shifted by every v supported on $\overline{\ell(T)}$. We refer to the set of all such trees $F = \{T \oplus v\}$ as an orchard, and we say that T belongs to the orchard F .

An orchard guarantees the existence of $2^{n-\ell(T)}$ trees that are isomorphic to T in G_f . It is *a priori* unclear that orchards exist in G_f . The following simple but key lemma proves their existence.

Lemma 4.6. Let T be a sensitive tree. Either T belongs to an orchard, or there exists a shift $T \oplus v$ of T which is not maximal.

Proof: Assume the tree T does not belong to an orchard. Pick the smallest weight vector v' supported on $\overline{\ell(T)}$ such that T cannot be shifted by v' (if there is more than one such vector any one will do). Since T can trivially be shifted by 0^n , we have $\text{wt}(v') \geq 1$. Pick any co-ordinate $i \in \text{supp}(v')$, and let $v = v' \oplus e_i$ so that $\text{wt}(v) = \text{wt}(v') - 1$. By our choice of v' , T can be shifted by v , but not by $v' = v \oplus e_i$. This implies that there exists $x \in V(T)$ so that $f(x) = f(x \oplus v) \neq f(x \oplus v')$, hence $T \oplus v$ is not maximal. \square

This lemma directly implies the existence of orchards for every G_f :

Corollary 4.7. *Every sensitive tree T where $|\ell(T)| = \text{ts}(f)$ belongs to an orchard.*

The lemma also gives the following intersection property for orchards. Since any two trees in an orchard F are isomorphic, we can define $\ell(F) = \ell(T)$ to be the set of edge labels for any tree $T \in F$.

Lemma 4.8. *Let F_1 and F_2 be orchards. Then $\ell(F_1) \cap \ell(F_2) \neq \emptyset$.*

Proof: Assume for contradiction that $\ell(F_1)$ and $\ell(F_2)$ are disjoint. We choose trees $T_1 \in F_1$ and $T_2 \in F_2$, and $x \in V(T_1), y \in V(T_2)$ such that $f(x) = 1$ and $f(y) = -1$. Now define $z \in \{0, 1\}^n$ where z_i equals x_i if $i \in \ell(T_1)$ and z_i equals y_i otherwise. Since z agrees with x on $\ell(T_1) = \ell(F_1)$, it can be obtained by shifting x by $z \oplus x$ which is supported on $\ell(T_1)$. Since T_1 belongs to an orchard, we get $f(z) = f(x) = 1$. However, we also have that $z_i = y_i$ for all $i \in \ell(T_2)$. Hence by similar reasoning, $f(z) = f(y) = -1$, which is a contradiction. \square

We use this intersection property to lower bound tree sensitivity in terms of decision tree depth, via an argument similar to other upper bounds on $\text{dt}(f)$ (such as the well known [BI87, Tar89, HH91] quadratic upper bound on $\text{dt}(f)$ in terms of certificate complexity).

Theorem 4.9. *For any Boolean function $f : \{0, 1\}^n \rightarrow \{\pm 1\}$, we have $\text{ts}(f) \geq \sqrt{2 \text{dt}(f)} - 1$.*

Proof: We construct a decision tree for f by iterating the following step until we are left with a constant function at each leaf: at the current node in the decision tree, pick the largest sensitive tree T in the (restricted) function and read all the variables in $\ell(T)$.

Let k be the largest number of iterations before we terminate, taken over all paths in the decision tree. Fix a path that achieves k iterations and let f_i be the restriction of f that is obtained, at the end of the i -th iteration (and let $f_0 = f$). We claim that $\text{ts}(f_i) \leq \text{ts}(f) - i$. Note that if f_i is not constant then $\text{ts}(f_i) \geq 1$, hence this claim implies that $k \leq \text{ts}(f)$.

It suffices to prove the case $i = 1$, since we can then apply the same argument repeatedly. Consider all trees in $f_0 = f$ of size $\text{ts}(f)$. Each of them occurs in an orchard by Corollary 4.7 and by Lemma 4.8 any two of them share at least one variable. Hence when we read all the variables in some tree T , we restrict at least one variable in every tree of size $\text{ts}(f)$, reducing the size by at least 1. The size of the other trees cannot increase after restriction, since G_{f_1} is an induced subgraph of G_f . Hence all the sensitive trees in f_1 have size at most $\text{ts}(f) - 1$.

It follows that overall we can bound the depth of the resulting decision tree by

$$\text{dt}(f) \leq \sum_{i=1}^k \text{ts}(f_{i-1}) \leq \sum_{i=1}^k (\text{ts}(f) - (i-1)) \leq \frac{\text{ts}(f)(\text{ts}(f) + 1)}{2}.$$

\square

It is natural to ask whether $\text{ts}(f)$ is polynomially related to $\text{dt}(f)$ and other standard complexity measures. Lemma 7.1 in Section 7 gives an example of a function on n variables where $\text{dt}(f) = \log(n+1)$ whereas $\text{ts}(f) = n$. In the other direction, it is likely that the bound in Theorem 4.9 can be improved further. We conjecture that the following bound should hold:

Conjecture 4.10. *For any Boolean function $f : \{0, 1\}^n \rightarrow \{\pm 1\}$, we have $\text{ts}(f) \geq \text{dt}(f)$ (and hence $\text{ts}(f) \geq \deg(f)$).*

In addition to being a natural question by itself, we will show in Section 4.3 that Conjecture 4.10 would have interesting consequences via the switching lemma in Section 4.2.

4.2 Tree Sensitivity under Random Restrictions

In this subsection we show that the probability of a random restriction of f having large tree sensitivity is both upper and lower bounded by suitable sensitivity moments of f .

Theorem 4.11. *Let $f : \{0, 1\}^n \rightarrow \{\pm 1\}$, $\rho \leftarrow \mathcal{R}_{k,n}$ and $1 \leq j \leq k$. Then we have*

$$\frac{s^j(f)}{n^j} \approx \frac{s^j(f)}{\prod_{i=0}^{j-1} (n-i)} \leq \Pr_{\rho \leftarrow \mathcal{R}_{k,n}} [\text{ts}(f_\rho) \geq j] \leq \frac{\binom{k}{j} 2^{k+2j} s^j(f)}{\binom{n}{j}} \approx 2^k \frac{(4k)^j s^j(f)}{n^j}.$$

The lower bound follows from the fact that $\text{ts}(f) \geq s(f)$ and Theorem 3.1. The key ingredient in the upper bound is Sidorenko's theorem [Sid94], which bounds the number of homomorphisms from a tree T with j edges to a graph G in terms of the j^{th} degree moment of G . For a formal statement of Sidorenko's theorems, we refer the reader to [Sid94, CL14]. Below, we state the result we will use in our language. We also present an elegant proof due to Levin and Peres [LP16] which seems considerably simpler than the known proofs of Sidorenko's theorem (though the lemma follows directly from that theorem).

Lemma 4.12. [LP16] *Let \mathcal{S}_j denote the set of sensitive trees of size j in G_f . Then we have that*

$$|\mathcal{S}_j| \leq 4^j \sum_{x \in \{0,1\}^n} s(f, x)^j.$$

Proof. We consider the set \mathcal{T} of all rooted unlabelled trees with j edges. It is known that $|\mathcal{T}| \leq 4^j$, indeed it equals the j^{th} Catalan number. Each tree $t \in \mathcal{T}$ has $j+1$ vertices. We label them $\{0, \dots, j\}$ where 0 is the root, and we label the remaining vertices using a breadth first search. This lets us define $p_t : \{1, \dots, j\} \rightarrow \{0, \dots, j-1\}$ where $p_t(i) < i$ is the parent of vertex i . Let $\mathcal{S}(t)$ denote the set of sensitive trees $T \in G_f$ whose adjacency structure is given by t .

For conciseness let us write $s_{\text{tot}}(f)$ to denote $\sum_{x \in \{0,1\}^n} s(f, x)$. Let \mathcal{D} denote the distribution on $\{0, 1\}^n$ where for each $x \in \{0, 1\}^n$,

$$\Pr_{\mathbf{x} \leftarrow \mathcal{D}}[\mathbf{x} = x] = \frac{s(f, x)}{s_{\text{tot}}(f)}.$$

Note that \mathcal{D} is supported only on vertices where $s(f, x) \geq 1$. Further \mathcal{D} is a stationary distribution for the simple random walk on G_f : if we sample a vertex from \mathcal{D} and then walk to a random neighbor, it is also distributed according to \mathcal{D} .

Fix a tree $t \in \mathcal{T}$ and consider a random walk on G_f which is the following vector $\mathbf{X} = (\mathbf{X}_0, \dots, \mathbf{X}_j)$ of random variables:

- We sample \mathbf{X}_0 from $\{0, 1\}^n$ according to \mathcal{D} .
- For $i \geq 1$, let \mathbf{X}_i be a random neighbor of $\mathbf{X}_{i'}$ in G_f where $i' = p_t(i) < i$.

Note that every \mathbf{X}_i is distributed according to \mathcal{D} . The vector $\mathbf{X} = (\mathbf{X}_0, \dots, \mathbf{X}_j)$ is such that $(\mathbf{X}_i, \mathbf{X}_{p_t(i)}) \in E(G_f)$, but it might contain repeated vertices and edge labels (indeed, this proof bounds the number of homomorphisms from G_f to t).

A vector $x = (x_0, \dots, x_j) \in (\{0, 1\}^n)^{j+1}$ will be sampled with probability

$$\begin{aligned}\Pr[\mathbf{X} = x] &= \Pr[\mathbf{X}_0 = x_0] \prod_{i=1}^j \Pr[\mathbf{X}_i = x_i | \mathbf{X}_0, \dots, \mathbf{X}_{i-1}] \\ &= \frac{s(f, x_0)}{\sum_{x \in \{0,1\}^n} s(f, x)} \prod_{i=1}^{j-1} \frac{1}{s(f, x_i)} \\ &= \frac{1}{\sum_{x \in \{0,1\}^n} s(f, x)} \prod_{i=1}^{j-1} \frac{1}{s(f, x_i)}.\end{aligned}$$

Clearly $\mathcal{S}(t)$ lies in the support of \mathbf{X} , hence

$$\begin{aligned}|\mathcal{S}(t)| &\leq \text{supp}(\mathbf{X}) \\ &\leq \mathbb{E}_{\mathbf{X}} \left[\frac{1}{\Pr[\mathbf{X} = x]} \right] \\ &\leq \mathbb{E}_{\mathbf{X}} \left[\sum_{x \in \{0,1\}^n} s(f, x) \prod_{i=1}^{j-1} s(f, \mathbf{X}_i) \right] \\ &= s_{\text{tot}}(f) \mathbb{E}_{\mathbf{X}} \left[\prod_{i=1}^{j-1} s(f, \mathbf{X}_i) \right] \\ &\leq s_{\text{tot}}(f) \mathbb{E}_{\mathbf{X}} \left[\frac{\sum_{i=1}^{j-1} s(f, \mathbf{X}_i)^{j-1}}{j-1} \right] \quad (\text{AM-GM Inequality}) \\ &= s_{\text{tot}}(f) \mathbb{E}_{\mathbf{Y} \sim \mathcal{D}} [s(f, \mathbf{Y})^{j-1}] \quad (8)\end{aligned}$$

where the last equality holds by linearity of expectation and the fact that all the \mathbf{X}_i 's are identically distributed. We bound the moment under \mathcal{D} as follows:

$$\begin{aligned}\mathbb{E}_{\mathbf{Y} \sim \mathcal{D}} [s(f, \mathbf{Y})^{j-1}] &\leq \sum_{y \in \{0,1\}^n} \Pr[\mathbf{Y} = y] s(f, y)^{j-1} \\ &= \sum_{y \in \{0,1\}^n} \frac{s(f, y)}{s_{\text{tot}}(f)} s(f, y)^{j-1} \\ &= \frac{\sum_{y \in \{0,1\}^n} s(f, y)^j}{s_{\text{tot}}(f)}.\end{aligned}$$

Plugging this back into Equation (8) gives

$$|\mathcal{S}(t)| \leq \sum_{y \in \{0,1\}^n} s(f, y)^j$$

Summing over all possibilities for t , we get

$$|\mathcal{S}_j| \leq \sum_{t \in \mathcal{T}} |\mathcal{S}(t)| \leq 4^j \sum_{y \in \{0,1\}^n} s(f, y)^j.$$

□

Theorem 4.11 now follows from an argument similar to Theorem 3.1.

Proof of Theorem 4.11. The lower bound follows from (the lower bound in) Theorem 3.1 and the observation that $\text{ts}(f_\rho) \geq s(f_\rho)$. We now prove the upper bound.

Similar to Theorem 3.1, consider the bipartite graph where the LHS is the set \mathcal{S}_j of all sensitive trees T of size j in G_f , the RHS is the set $\mathcal{R}_{k,n}$ of all restrictions ρ , and (T, ρ) is an edge if the tree T lies in the subcube $C(\rho)$ specified by the restriction ρ . The desired probability $\Pr_{\rho \in \mathcal{R}_{k,n}}[\text{ts}(f_\rho) \geq j]$ is the fraction of nodes in $\mathcal{R}_{k,n}$ that are incident to at least one edge.

We first bound the degree of each vertex on the left. To have T lying in $C(\rho)$,

- The edge labels of T must be live variables for ρ .
- The values ρ_i for the fixed coordinates $i \in [n] \setminus L(\rho)$ must be consistent with the values in $V(T)$.

The only choice is of the $(k - j)$ remaining live coordinates. Hence $T \in C(\rho)$ for at most $\binom{n-j}{k-j}$ values of ρ corresponding to choices of the remaining live variables.

The number of vertices in \mathcal{S}_j is bounded using Lemma 4.12 by

$$|\mathcal{S}_j| \leq 4^j \sum_{x \in \{0,1\}^n} s(f, x)^j = 4^j 2^n s^j(f),$$

so the total number of edges is at most

$$\binom{n-j}{k-j} 2^n 4^j s^j(f).$$

A restriction $\rho \in \mathcal{R}_{k,n}$ is specified by a set $L(\rho)$ of k live co-ordinates where $\rho_i = \star$, and a value $\rho_i \in \{0, 1\}$ for the other coordinates, and hence

$$|\mathcal{R}_{k,n}| = \binom{n}{k} 2^{n-k}.$$

Recall that $\text{ts}(f_\rho) \geq j$ iff $C(\rho)$ contains some tree from \mathcal{S}_j . Hence the fraction of restrictions ρ that have an edge incident to them is

$$\Pr_{\rho \in \mathcal{R}_{k,n}}[\text{ts}(f_\rho) \geq j] \leq \frac{\binom{n-j}{k-j} 2^n 4^j s^j(f)}{\binom{n}{k} 2^{n-k}} \leq \frac{\binom{k}{j} 2^{k+2j} s^j(f)}{\binom{n}{j}}.$$

□

4.3 An approach to Conjecture 1.3

By combining Theorems 3.1, 4.9 and 4.11, we get upper and lower bounds on the probability that a random restriction of a function has large decision tree depth in terms of its sensitivity moments.

Corollary 4.13. *Let $f : \{0, 1\}^n \rightarrow \{\pm 1\}$, $\rho \sim \mathcal{R}_{k,n}$ and $1 \leq j \leq k$. Then*

$$\frac{s^j(f)}{n^j} \approx \frac{s^j(f)}{\prod_{i=0}^{j-1} (n-i)} \leq \Pr_{\rho \leftarrow \mathcal{R}_{k,n}}[\text{dt}(f_\rho) \geq j] \leq 8^k \frac{\binom{k}{\sqrt{2j}-1} s^{\sqrt{2j}-1}(f)}{n^{\sqrt{2j}-1}} \approx \frac{8^k k^{\sqrt{2j}-1} s^{\sqrt{2j}-1}(f)}{n^{\sqrt{2j}-1}}.$$

Note that the denominator in the lower bound is $n^{\Omega(j)}$ but for the upper bound, it is $n^{\Omega(\sqrt{j})}$. This quadratic gap comes from Theorem 4.9. However, if Conjecture 4.10 stating that $\text{ts}(f) \geq \text{dt}(f)$ were true, it would imply the following sharper upper bound.

Corollary 4.14. *Let $f : \{0, 1\}^n \rightarrow \{\pm 1\}$, $\rho \sim \mathcal{R}_{k,n}$ and $1 \leq j \leq k$. If Conjecture 4.10 holds (in the stronger form that $\text{ts}(f) \geq \text{dt}(f)$), then*

$$\Pr_{\rho \leftarrow \mathcal{R}_{k,n}} [\text{dt}(f_\rho) \geq j] \leq \frac{\binom{k}{j} 2^{k+2j} s^j(f)}{\binom{n}{j}},$$

and if Conjecture 4.10 holds in the weaker form that $\text{ts}(f) \geq \deg(f)$, then

$$\Pr_{\rho \leftarrow \mathcal{R}_{k,n}} [\deg(f_\rho) \geq j] \leq \frac{\binom{k}{j} 2^{k+2j} s^j(f)}{\binom{n}{j}}.$$

The dependence on n here matches that in the lower bound of Corollary 4.13. Conjecture 1.3 follows from this as an easy consequence.

Corollary 4.15. *Conjecture 4.10 (in fact, the weaker form that $\text{ts}(f) \geq \deg(f)$) implies Conjecture 1.3.*

Proof: We will prove that $\mathbb{I}^k(f) \leq 8^k k! s^k(f)$. Let $\rho \leftarrow \mathcal{R}_{k,n}$ and consider the event that $\deg(f_\rho) = k$. By Theorem 3.2, we can lower bound this probability in terms of the Fourier moments of f as

$$\frac{\mathbb{I}^k(f)}{\prod_{i=0}^{k-1} (n-i)} \leq \Pr_{\rho \leftarrow \mathcal{R}_{k,n}} [\deg(f_\rho) = k].$$

To upper bound it, by Corollary 4.14, if the weaker form of Conjecture 4.10 holds, then we have

$$\Pr_{\rho \leftarrow \mathcal{R}_{k,n}} [\deg(f_\rho) \geq k] \leq \frac{2^{3k} s^k(f)}{\binom{n}{k}}.$$

The claim follows by comparing the upper and lower bounds. \square

For $k = 3, 4$, it is an easy exercise to verify that $\deg(f_\rho) = k$ implies $\text{ts}(f_\rho) \geq k$. This implies that Conjecture 1.3 holds for $k = 3, 4$.

5 Fourier concentration for low-sensitivity functions

5.1 Proper Walks

Since $s^j(f) \leq (s(f))^j$ for all j , one can trivially bound the sensitivity moments of a function in terms of its max sensitivity. Hence Corollaries 4.14 and 4.15 show that under Conjecture 4.10, low sensitivity functions simplify under random restrictions. In this section we prove this unconditionally. The key ingredient is a relaxation of sensitive trees that we call *proper walks*.

A *walk* W in the n -dimensional Boolean cube is a sequence of vertices (w_0, w_1, \dots, w_t) such that w_i and w_{i+1} are at Hamming distance precisely 1. We allow walk to backtrack and visit vertices more than once. We say that t is the *length* of such a walk.

Let $\ell(W) \subseteq [n]$ denote the set of coordinates that are flipped by walk W . We define $k = |\ell(W)|$ to be the *dimension* of the walk. We order the coordinates in $\ell(W)$ as ℓ_1, \dots, ℓ_k according to the order in which they are first flipped. For each $\ell_i \in \ell(W)$, let x_i denote the first vertex in W at which we flip coordinate i .

Definition 5.1. A walk W is a proper walk for a Boolean function $f : \{0, 1\}^n \rightarrow \{\pm 1\}$ if for each $\ell_i \in \ell(W)$, the vertex x_i is sensitive to ℓ_i .

Thus a walk is proper for f if the first edge flipped along a new coordinate direction is always sensitive. This implies that while walking from x_i to x_{i+1} , we are only allowed to flip a subset of the coordinates $\{\ell_1, \dots, \ell_i\}$, hence $\text{supp}(x_i \oplus x_{i+1}) \subseteq \{\ell_1, \dots, \ell_i\}$. Hence if there is a proper walk of dimension k then there is one of length at most $k(k+1)/2$, by choosing a shortest path between x_i and x_{i+1} for each i .

In studying proper walks, it is natural to try to maximize the dimension and minimize the length. We first focus on the former. The following lemma states that the obvious necessary condition for the existence of an n -dimensional walk is in fact also sufficient:

Lemma 5.2. Every Boolean function $f : \{0, 1\}^n \rightarrow \{\pm 1\}$ that depends on all n coordinates has a proper walk of dimension n .

Proof: Pick $\ell_1 \in [n]$ arbitrarily and let x_1 be any vertex in $\{0, 1\}^n$ which is sensitive to coordinate ℓ_1 . Let $1 \leq i \leq n$. Inductively we assume we have picked coordinates $L = \{\ell_1, \dots, \ell_i\}$ and points $X = \{x_1, \dots, x_i\}$ so that for every $j \leq i$,

1. x_j is sensitive to ℓ_j .
2. For $j \geq 2$, $\text{supp}(x_{j-1} \oplus x_j) \subseteq \{\ell_1, \dots, \ell_{j-1}\}$.

If we visit x_1, \dots, x_i in that order and walk from each x_j to x_{j+1} along a shortest path, the resulting walk is a proper walk for f . Let C be the subcube that spans the dimensions in L and contains X .

Case 1: Some vertex in C is sensitive to a coordinate outside of L . Name this vertex x_{i+1} and the sensitive co-ordinate ℓ_{i+1} , and add them to X and L respectively. Note that $x_i \oplus x_{i+1}$ is indeed supported on $\{\ell_1, \dots, \ell_i\}$, so both conditions (1) and (2) are met.

Case 2: No vertex in C is sensitive to a coordinate outside L . So for any co-ordinate $j \notin L$, we have $f(x) = f(x \oplus e_j)$. But this means that the set of points $X \oplus e_j$ and co-ordinates L also satisfy the inductive hypothesis (specifically conditions (1) and (2) above).

Let d denote the Hamming distance from C to the closest vertex which is sensitive to some coordinate outside L . Let z denote one such closest vertex to C (there could be many) and pick any coordinate j in which z differs from the closest point in C . If we replace X by $X \oplus e_j$, the Hamming distance to z has decreased to $d-1$. We can repeat this till the Hamming distance drops to 0, which puts us in Case (1). \square

Given this result, it is natural to try to find full dimensional walks of the smallest possible length. The length of the walk constructed above is bounded by $\sum_{i=1}^n (i-1) \leq n^2/2$. Lemma 7.2 in Section 7 gives an example showing that this is tight up to constants. So while we cannot improve the bound in general, we are interested in the case of functions with large decision tree complexity, where the following observation suggests that better bounds should be possible.

Lemma 5.3. If $\text{ts}(f) = n$, then f has a proper walk of dimension n and length $2n$.

The proof is by performing a traversal on a sensitive tree of dimension n , starting and ending at the root, going over each edge twice. Thus if Conjecture 4.10 were true, it would imply that functions requiring full decision tree depth have proper walks of length $O(n)$. We now give an unconditional proof of this result (we will use it as an essential ingredient in our “switching lemma” later).

Theorem 5.4. *If $\text{dt}(f) = n$, then f has a proper walk of dimension n and length at most $3n$.*

Proof: The proof is by induction on n . The base case $n = 2$ is trivial since in this case there exists a proper walk of length 2. Assume the claim holds for all $n' < n$. Let f be a function where $\text{dt}(f) = n$. If $\text{ts}(f) = n$ we are done by Lemma 5.3, so we assume that $\text{ts}(f) = m < n$. By Corollary 4.7, there is an orchard $\{T \oplus v\}$ of sensitive trees where $\dim(T) = m$. Assume by relabeling that $\ell(T) = \{1, \dots, m\}$.

Since $\text{dt}(f) = n$, there exists a setting t_1, \dots, t_m of variables in $[m]$ such that the restriction $f' = f|_{x_1=t_1, \dots, x_m=t_m}$ on $n' = n - m$ variables satisfies $\text{dt}(f') = n - m$. By the inductive hypothesis, there exists a proper walk in f' of dimension $n - m$ and length $3(n - m)$ in the subcube $x_1 = t_1, \dots, x_m = t_m$ which starts at some vertex $s' = (t_1, \dots, t_m, s'_{m+1}, \dots, s'_n)$ and ends at some vertex $t' = (t_1, \dots, t_m, t'_{m+1}, \dots, t'_n)$, which flips all coordinates in $[n] \setminus [m]$.

Consider the tree $T \oplus v$ in the orchard such that the coordinates of $V(T \oplus v)$ in $[n] \setminus [m]$ agree with s' . Our walk can be divided into three phases:

1. By Lemma 5.3, we can visit every vertex in $T \oplus v$ using a proper walk of length $2m$ that only uses edges in $[m]$. Assume that this walk starts and ends at r . By our choice of v we have that $(r_{m+1}, \dots, r_n) = (s'_{m+1}, \dots, s'_n)$.
2. From r , we then walk to the vertex $s' = (t_1, \dots, t_m, s'_{m+1}, \dots, s'_n)$. This only requires flipping bits in $[m]$, so it keeps the walk proper and adds only at most m to its length.
3. The inductive hypothesis applied to f' allows us to construct a proper walk from s' to t' that only walks along edges in $[n] \setminus [m]$ and has length at most $3(n - m)$.

Thus the total length of the walk is at most $2m + m + 3(n - m) = 3n$. \square

5.2 Random restrictions of low sensitivity functions

In this section we prove our “switching lemma for low-sensitivity functions”, Lemma 5.5. The high-level idea is to count the number of (short) proper walks that arise when using Theorem 5.4. This allows us to upper bound the number of restrictions ρ for which f_ρ has full decision tree (since each such restriction yields a short proper walk by Theorem 5.4). We follow Razborov’s proof strategy for the switching lemma [Raz95]. In order to bound the number of walks, we encode each walk using a short description and then show that this encoding is a bijection. Since the encoding is bijective, the number of walks is at most the number of encodings and we get the required upper bound.

Lemma 5.5. *Let $f : \{0, 1\}^n \rightarrow \{\pm 1\}$. Then*

$$\Pr_{\rho \leftarrow \mathcal{R}_{k,n}} [\text{dt}(f_\rho) = k] \leq \frac{(32s(f))^k}{\binom{n}{k}}.$$

Proof: We will prove the Lemma via an encoding scheme. Let $S = \{\rho \in \mathcal{R}_{k,n} : \text{dt}(f_\rho) = k\}$. We shall prove that $|S| \leq 2^n \cdot (16s(f))^k$. This will complete the proof since

$$\Pr_{\rho \leftarrow \mathcal{R}_{k,n}} [\text{dt}(f_\rho) = k] = \frac{|S|}{|\mathcal{R}_{k,n}|} = \frac{2^n \cdot (16s(f))^k}{2^{n-k} \cdot \binom{n}{k}} = \frac{(32s(f))^k}{\binom{n}{k}}.$$

For shorthand, let $s = s(f)$. We define an encoding

$$E : S \rightarrow \{0, 1\}^n \times \{0, 1\}^k \times \{0, 1\}^{2k} \times \{0, 1\}^k \times [s]^k,$$

and show that E is a bijection, hence $|S| \leq 2^n \cdot 2^{4k} \cdot s^k$. Given a restriction $\rho \in S$, let $W = v_0, v_1, \dots, v_{3k}$ be the proper walk defined in the proof of Theorem 5.4. We encode W to $(v_0, 1_K, b, c, \beta)$ where $K \subseteq [k]$, $b \in \{0, 1\}^{2k}$, $c \in \{0, 1\}^k$ and $\beta \in [s]^k$ are defined next.

We open up the recursive argument from Theorem 5.4. The walk W consists of $t \leq k$ phases. Each phase $i \in [t]$ consists of a traversal over a sensitive tree T_i with k_i edges, and then a walk to a point in the minimal cube containing T_i . The root of T_{i+1} is the end point of the walk of phase i .

Encoding the tree sizes. We encode using a binary string of length k the numbers (k_1, \dots, k_t) . This can be done by letting $K = \{k'_1, k'_2, \dots, k'_t\}$ where $k'_i = k_1 + \dots + k_i$ and taking the indicator vector of K . Since all k_i -s are nonnegative and $k_1 + \dots + k_t = k$, we get that all k'_1, \dots, k'_t are distinct numbers between 1 and k . Thus, the set K is a subset of $[k]$ and may be encoded using k bits.

Encoding each tree. In phase $i \in \{1, \dots, t\}$, given the initial node of the phase, which is the root of T_i , denoted r_i , we show how to encode T_i using $2k_i$ bits in addition to k_i numbers in $[s]$ (recall that k_i is the number of edges in T_i).

Take a traversal over the tree T_i starting at the root and finishing at the root. The length of this walk is $2k_i$ since we go on each edge once in each direction. We encode the walk as a binary string of length $2k_i$ in addition to a sequence in $[s]^{k_i}$. For the walk $r_i = u_0, u_1, \dots, u_{2k_i}$, define the encoding $(b^{(i)}, \beta^{(i)}) \in \{0, 1\}^{2k_i} \times [s]^{k_i}$ as follows. Initialize $\beta^{(i)}$ to be the empty sequence, and $b^{(i)}$ to be the all zeros sequence 0^{k_i} . For $j = 1, \dots, 2k_i$, either u_j is the father of u_{j-1} in the sensitive tree, or u_j is a child of u_{j-1} . Put $b_j^{(i)} = 1$ iff u_j is a child of u_{j-1} . In such a case, let $n_j \in [s]$ be the index of u_j among all sensitive neighbors of u_{j-1} in the hypercube, according to some canonical order on $\{0, 1\}^n$, and append n_j to the sequence $\beta^{(i)}$.

After finishing the walk, let $\ell_1 < \ell_2 \dots < \ell_{k_i}$ be the coordinates that the sensitive tree T_i changes (i.e., $\ell(T_i)$). We encode using $c^{(i)} \in \{0, 1\}^{k_i}$ the walk inside the minimal cube containing T_i by setting $c_j^{(i)} = 1$ iff the ℓ_j coordinate should be flipped during the walk, for $j = 1, \dots, k_i$.

Finally, after finishing the t phases, we let $b = b^{(1)} \circ \dots \circ b^{(t)}$, $\beta = \beta^{(1)} \circ \dots \circ \beta^{(t)}$ and $c = c^{(1)} \circ \dots \circ c^{(t)}$ be the concatenation of the substrings defined for each phase.

Decoding. We show that a decoder can uniquely recover the proper walk from the encoding. This will allow us to show that E is a bijection. First, the decoder knows k_1, \dots, k_t (and t) since it can decode the set $K = \{k'_1, \dots, k'_t\}$ that determines k_1, \dots, k_t by $k_1 = k'_1$ and $k_i = k'_i - k'_{i-1}$ for $i = 2, \dots, t$.

For $i = 1, \dots, t$ we show that assuming the decoder knows the starting position of the walk in phase i , i.e., r_i , it decodes phase i successfully. Since by the end of phase i we reach r_{i+1} this shows that the entire decoding procedure works.

Given the tree sizes k_1, \dots, k_t , the decoder may identify the substrings $b^{(i)} \in \{0, 1\}^{2k_i}$, $\beta^{(i)} \in [s]^{k_i}$ and $c^{(i)} \in \{0, 1\}^{k_i}$ inside b , β and c respectively. To follow the traversal in the sensitive tree T_i the decoder reads bits from $b^{(i)}$ indicating whether one should go to a child of the current node or move back to its parent. In the case that the traversal goes to a child of the current node, the next

symbol from $\beta^{(i)}$ is read, indicating which one of the sensitive neighbors of the current node is this child.

After reading $2k_i$ bits from b and k_i symbols from β , the decoder finishes reconstructing the tree traversal of T_i . Thus, it identifies correctly the coordinates $\ell_1 < \dots < \ell_{k_i}$ as the sensitive coordinates in the tree T_i . Next, the decoder walks along the path defined by $c^{(i)}$, flipping the ℓ_j coordinate iff $c_j^{(i)} = 1$ for $j = 1, \dots, k_i$. This completes the decoding of phase i .

E is a bijection. Let $\rho \in S$. Given $E(\rho) = (x_0, 1_K, b, c, \beta)$ the decoder finds a proper walk W with $|\ell(W)| = k$, that is contained in the subcube defined by ρ . Thus, the minimal subcube containing the walk W uniquely determines the restriction ρ , and we get that E is bijective. \square

We note that one can prove a similar bound for $\Pr_{\rho \in \mathcal{R}_{k,n}}[\text{dt}(f_\rho) \geq j]$; here we have presented only the case $j = k$ both because it is simpler and because it suffices for the concentration results in Section 5.3.

We would like to replace the $(s(f))^k$ term with $s^k(f)$, the k^{th} sensitivity moment. The above proof does not seem to generalize to that case, because we do not have an analogue of Sidorenko's result on trees for proper walks.

5.3 Fourier tails of low sensitivity functions

We have the necessary pieces in place to give an upper bound on $\mathbb{I}^k[f]$:

Lemma 5.6. *For every $f : \{0, 1\}^n \rightarrow \{\pm 1\}$ and every $k \geq 1$, we have $\mathbb{I}^k[f] \leq (32s(f))^k \cdot k!$*

Proof. By Theorem 3.2 and Lemma 5.5, we have that

$$\frac{\mathbb{I}^k[f]}{\prod_{i=0}^{k-1} (n-i)} \leq \Pr_{\rho \leftarrow \mathcal{R}_{k,n}} [\deg(f_\rho) = k] \leq \Pr_{\rho \leftarrow \mathcal{R}_{k,n}} [\text{dt}(f_\rho) = k] \leq \frac{(32s(f))^k}{\binom{n}{k}},$$

which may be rewritten as the claimed bound. \square

Now we are ready to prove Theorem 1.2:

Theorem 1.2. *For any function f and any $\epsilon > 0$, we have $\deg_\epsilon(f) \leq O(s(f) \cdot \log(1/\epsilon))$.*

Proof. Let $k = \log(1/\epsilon)$, and let $t \geq k$ be some parameter to be determined later. We have

$$\sum_{|S| \geq t} \hat{f}(S)^2 = \Pr_{\mathbf{R} \leftarrow \mathcal{D}_f} [|\mathbf{R}| \geq t].$$

Since $\binom{t}{k}$ is strictly increasing for $t \geq k$, we have

$$\Pr_{\mathbf{R} \leftarrow \mathcal{D}_f} [|\mathbf{R}| \geq t] = \Pr_{\mathbf{R} \leftarrow \mathcal{D}_f} \left[\binom{|\mathbf{R}|}{k} \geq \binom{t}{k} \right].$$

Now observe that we have

$$\frac{\mathbb{I}^k[f]}{k!} = \mathbb{E}_{\mathbf{R} \leftarrow \mathcal{D}_f} \left[\binom{|\mathbf{R}|}{k} \right].$$

Hence Markov's inequality and Lemma 5.6 gives

$$\Pr_{\mathbf{R} \leftarrow \mathcal{D}_f} [|\mathbf{R}| \geq t] \leq \frac{\mathbb{E}_{\mathbf{R} \leftarrow \mathcal{D}_f} [\binom{|\mathbf{R}|}{k}]}{\binom{t}{k}} \leq \frac{\mathbb{E}[f]/k!}{(t/k)^k} \leq \frac{(32s(f))^k}{(t/k)^k}.$$

Choosing $t = 64s(f) \cdot k$ gives

$$\Pr_{\mathbf{R} \leftarrow \mathcal{D}_f} [|\mathbf{R}| \geq t] \leq \frac{1}{2^k} \leq \epsilon.$$

Overall, we have

$$\deg_\epsilon(f) \leq t = O(s(f) \cdot \log(1/\epsilon)).$$

□

We note that the relations between influence moments and Fourier concentration that are established in [Tal14b, Section 4] can also be used to obtain Theorem 1.2 from Lemma 5.6. That work [Tal14b, Section 4] also shows that bounded k -th influence moments imply bounded Fourier L_1 spectral norm on the k -th level, which in turn implies Fourier concentration on a small number of Fourier coefficients (smaller than the trivial $\binom{n}{k}$ bound on the number of coefficients at degree k). These results can be used with Lemma 5.6 to establish that functions with bounded max sensitivity have sparse Fourier spectra.

Corollary 5.7. *Let f be a Boolean function with sensitivity s . For some absolute constant c ,*

- $\sum_{S: |S|=k} |\hat{f}(S)| \leq (cs)^k$.
- f can be ϵ -approximated in L_2 by a polynomial with at most $s^{c \cdot s \cdot \log(1/\epsilon)}$ monomials.

5.4 On the tightness of the Fourier concentration bound

Recall that Conjecture 1.1 asserts the existence of $c > 0$ such that $\deg(f) \leq s(f)^c$, where c needs to be at least 2. In contrast, we have shown that $\deg_\epsilon(f) \leq s(f) \log(1/\epsilon)$. A possible approach to the Conjecture might be to tradeoff between the exponents of s and $\log(1/\epsilon)$, by showing bounds of the form $\deg_\epsilon(f) = O(s(f)^c \cdot \log(1/\epsilon)^\gamma)$ for $c \geq 1$ and $\gamma < 1$. We prove two claims about such improvements.

1. Any bound with constants (c, γ) where $\gamma < 1$ will imply Conjecture 4.10. Concretely, we will show if $\deg_\epsilon(f) = O(s(f)^c \cdot \log(1/\epsilon)^\gamma)$ for all $\epsilon > 0$ and $\gamma < 1$, then Conjecture 1.1 holds.
2. For a bound to hold with constants (c, γ) where $\gamma < 1$, we would need to have $c > 1$. Concretely, for any positive integer s and any $\epsilon \geq \Omega(1/s^s)$ there exists a Boolean function with sensitivity at most s and $\deg_\epsilon(f) = \Omega(s \cdot \frac{\log 1/\epsilon}{\log \log 1/\epsilon})$.

Lemma 5.8. *If there exist constants $c > 0$ and $\gamma < 1$ such that for all $\epsilon > 0$, $\deg_\epsilon(f) \leq O(s(f)^c \log(1/\epsilon)^\gamma)$, then $\deg(f) \leq O(s(f)^{c/(1-\gamma)})$ hence Conjecture 1.1 holds true.*

Proof: Let d denote $\deg(f)$ and take $\epsilon = 2^{-3d}$. For ϵ so small, we have $\deg_\epsilon(f) = \deg(f) = d$ as any Fourier coefficient is of magnitude at least 2^{-d} and the Fourier tail could be either 0 or at least 2^{-2d} . Thus, we would get

$$d = \deg_\epsilon(f) \leq O(s(f)^c \cdot \log(1/\epsilon)^\gamma) \leq O(s(f)^c \cdot (3d)^\gamma)$$

which is equivalent to $d = O(s(f)^{c/(1-\gamma)})$.

□

For the second claim, we give an example showing that our Fourier-concentration result is essentially tight. Our construction is based on the Hamming code. The proof is in Section 7.

Lemma 5.9. *Let s be a positive integer. For every $\epsilon \geq 0.5 \cdot s^{-s}$, there exists a Boolean function with sensitivity at most s such that $\deg_\epsilon(f) = \Omega(s \cdot \frac{\log(1/\epsilon)}{\log \log(1/\epsilon)})$.*

6 Applications

6.1 Learning Low-Sensitivity Functions

In Valiant’s PAC learning model, a learning algorithm tries to learn an unknown concept (function) from a class using a hypothesis function. While hypotheses are easy to model using low-level circuit classes, the question of modeling real-world concept classes accurately is more challenging, and has received a lot of attention in the machine learning literature. A fairly standard assumption is that concepts to be learned satisfy some kind of smoothness: roughly speaking, one often assumes that x and y being sufficiently close *generally implies* that $f(x) \approx f(y)$ [BL07, BCV13]. This assumption favors local prediction algorithms where the prediction of $f(x)$ is dominated by the values at near neighbors of x . While we have not found a completely formal definition, [BL07] call f smooth *when the value of f and its derivative f' at x and y are close whenever x and y are close*. This requires a suitable metric on both the domain and the range of f . It is natural to ask what smoothness means for Boolean functions, and how the assumption of smoothness affects the complexity of computing and learning a function f .

The work of [GNS⁺16] proposed low worst-case sensitivity as a notion of smoothness for Boolean functions, motivated by connections to classical analysis; we refer the reader to that paper for more details. We believe that low sensitivity is indeed a realistic assumption for several learning tasks (for instance, the concepts learned in image recognition tasks are arguably not sensitive to most individual pixels). For the natural definition of the gradient ∇f of f , [GSW16b] show that low sensitivity of f implies that the gradient also has low sensitivity, meaning that a random neighbor of x is likely to be sensitive to the same set of coordinates as the point x itself. Thus low-sensitivity functions have the property that f and ∇f are likely to agree at nearby points.

Relevant work along these lines is that of Klivans et al. [KOS04] on learning noise-stable Boolean functions. One can view low noise-sensitivity under the uniform distribution on inputs as a notion of smoothness which guarantees that nearby points are likely to have similar values. [KOS04] showed that low (uniform-distribution) noise-sensitivity implies efficient learnability under the uniform distribution, since it implies Fourier concentration. Low sensitivity is a stronger assumption than low noise-sensitivity; [GNS⁺16, Lemma 15] shows that it implies a strong *pointwise* noise stability bound, which says that for any x , most y which are close to it satisfy $f(y) = f(x)$. In contrast, in the definition of noise-sensitivity both x and y are chosen randomly.

Conjecture 1.1 implies that low-sensitivity functions in fact have strong global structure: they have low polynomial degree and small depth as decision trees, so they can be PAC-learned efficiently under arbitrary distributions in time $O(n^{\text{poly}(s(f))})$. The best provable bound we are aware of is $n^{\exp(O(s))}$, which follows from Simon’s result [Sim82] that any sensitivity- s function depends on at most $\exp(O(s))$ variables. In this section, we give an efficient algorithm for learning low-sensitivity functions under the uniform distribution from random examples, and in fact show that exact learning is possible in this model.

We consider the problem of learning a concept f from a class of functions \mathcal{C} given uniformly random examples labeled according to f . In the standard PAC learning scenario, the goal is produce a hypothesis h such that with high probability (over the coin tosses of the algorithm and the random examples), $\Pr_{x \in \{0,1\}^n}[f(x) = h(x)] \geq 1 - \epsilon$. We refer to this as learning under the uniform distribution with error ϵ . In the exact learning problem ϵ is taken to be zero; the algorithm is allowed to fail with small probability, but otherwise it must return (an efficiently evaluable representation of) a hypothesis h that agrees with f on every $x \in \{0,1\}^n$.

The seminal work of [LMN93] showed that Fourier concentration bounds lead to uniform distribution learning algorithms. Applying their result with Theorem 1.2 we get

Theorem 6.1. *There is an algorithm that learns the class of sensitivity s functions on n variables under the uniform distribution with error ϵ in time $n^{O(s \log(1/\epsilon))}$.*

The algorithm runs the low-degree algorithm of [LMN93] to estimate the Fourier spectrum on the bottom $O(s \log(1/\epsilon))$ levels. It then returns the sign of the resulting polynomial as its hypothesis. This is already a stronger bound than one obtains by assuming low noise-sensitivity (see [KOS04, Corollary 17]), since those results have a polynomial in $1/\epsilon$ in the exponent of n . Further, we can use this to get an exact learning algorithm under the uniform distribution, using the following self-correction result from [GNS⁺16].

Theorem 6.2. [GNS⁺16, Theorem 4] *There exists a constant C such that given $r : \{0,1\}^n \rightarrow \{0,1\}$ such that $\Pr_{x \in \{0,1\}^n}[r(x) \neq f(x)] \leq 2^{-Cs}$ for some sensitivity s function f , there is a self-correction algorithm which when given an oracle for r and $x \in \{0,1\}^n$ as input, queries the oracle for r at $n^{O(s)}$ points, runs in $n^{O(s)}$ time, and returns $f(x)$ with probability 0.99.*

We can use this result to give an exact learning algorithm that outputs a randomized Boolean function as a hypothesis. Let us define precisely what this means. A randomized Boolean function h outputs a Boolean random variable $h(x)$ for each input $x \in \{0,1\}^n$, by tossing some random coins and performing a randomized computation. Our learning algorithm uses the examples to output a randomized Boolean function h as a hypothesis. With high probability over the examples, the resulting hypothesis h will have the guarantee that for every x , $\Pr_h[h(x) = f(x)] \geq 0.99$ where this probability is only over the coin tosses h uses to evaluate $h(x)$.

Theorem 6.3. *There is an algorithm for exactly learning the class of sensitivity s functions on n variables under the uniform distribution in time $n^{O(s^2)}$. The output hypothesis h is a randomized Boolean function which requires time $n^{O(s)}$ to evaluate on each input and which satisfies $\Pr_h[h(x) = f(x)] \geq 0.99$ for every x .*

Proof: We first use Theorem 6.1 with $\epsilon = 2^{-Cs}$ where C is the constant from Theorem 6.2 to get a learn a hypothesis r such that $\Pr_{x \in \{0,1\}^n}[r(x) \neq f(x)] \leq 2^{-Cs}$ in time $n^{O(s^2)}$, and invoke the self-corrector from Theorem 6.2 on r . Let h denote the randomized Boolean function which is the composition of the self-corrector with r . By Theorem 6.2, h has the properties claimed above. \square

The hypothesis above is a randomized Boolean function, which may err with some probability on any input. We can eliminate this error and with high probability learn a hypothesis that is correct on every input, at the price of a small increase in the running time. This requires the following result from [GNS⁺16].

Theorem 6.4. [*GNS⁺16*, Theorem 3] Let $f : \{0,1\}^n \rightarrow \{\pm 1\}$ be a sensitivity s function. Given the values of f on any Hamming ball of radius $2s$ as advice, there is an algorithm which, given $x \in \{0,1\}^n$ as input, runs in time $n^{O(s)}$ and computes $f(x)$.

Theorem 6.5. There is an algorithm that exactly learns the class of sensitivity s functions on n variables under the uniform distribution in time $n^{O(s^2 \log(n))}$. The output hypothesis is Boolean function which requires time $n^{O(s)}$ to evaluate on each input and which equals f on every input in $\{0,1\}^n$ with high probability over the learning examples.

Proof: We first use Theorem 6.1 with $\epsilon = 1/n^{3s}$. The LMN algorithm runs in time $n^{O(s^2 \log(n))}$ and with high probability returns a hypothesis h such that

$$\Pr_{\mathbf{x} \in \{0,1\}^n} [f(\mathbf{x}) \neq h(\mathbf{x})] \leq 1/n^{3s}.$$

We then pick a random ball \mathcal{B} of radius $2s$, and use h to label all the points in it. Since each point in the ball is uniformly random, all the labels will be correct with probability $1 - 1/n^s$. We then use these as advice for the algorithm in Theorem 6.4. On an input $x \in \{0,1\}^n$, the resulting hypothesis can be evaluated in time $n^{O(s)}$ and return the correct value $f(x)$. \square

6.2 The Entropy-Influence Conjecture revisited

In this section, we prove a bound on the entropy of the Fourier spectrum of f , in terms of the influence of f and its sensitivity. We begin with the definition of the entropy of the Fourier spectrum of f .

Definition 6.6. The Fourier-entropy of a Boolean function $f : \{0,1\}^n \rightarrow \{\pm 1\}$ is defined to be

$$\mathbb{H}[f] \stackrel{\text{def}}{=} \sum_{S \subseteq [n]} \hat{f}(S)^2 \cdot \log_2 \left(\frac{1}{\hat{f}(S)^2} \right)$$

The Fourier Entropy Influence Conjecture by Kalai and Friedgut [FK96] states that for any Boolean function f ,

$$\mathbb{H}[f] = O(\mathbb{I}[f]).$$

We upper bound the Fourier-entropy $\mathbb{H}[f]$ as a function of the influence and the sensitivity of f .

Theorem 6.7. For any Boolean function $f : \{0,1\}^n \rightarrow \{\pm 1\}$,

$$\mathbb{H}[f] \leq \mathbb{I}[f] \cdot (2 \log s[f] + O(1)).$$

This improves on the bound $\mathbb{H}[f] \leq \mathbb{I}[f] \cdot O(\log n + 1)$ given by O'Donnell, Wright and Zhou [OWZ11] (who also deduced better bounds for symmetric and block-symmetric functions).

In the remainder of this section, we shall denote by $\mathbb{W}^k[f]$ the sum $\sum_{S: |S|=k} \hat{f}(S)^2$. We use the following Theorem of O'Donnell et al. [OWZ11].

Theorem 6.8 ([OWZ11, Theorem 5]). Let $f : \{0,1\}^n \rightarrow \{\pm 1\}$ be a Boolean function. Then $\sum_{k=0}^n \mathbb{W}^k[f] \cdot \log \frac{1}{\mathbb{W}^k[f]} \leq 3 \cdot \mathbb{I}[f]$.

We use the following upper bound on the entropy of a distribution. To be slightly more general we state the inequality for all sequences of non-negative numbers and not just for those that sum up to 1.

Lemma 6.9. *Let (p_1, \dots, p_m) be a sequence of non-negative numbers and let $p = \sum_{i=1}^m p_i$. Then,*

$$\sum_{i=1}^m p_i \log(1/p_i) \leq 2p \cdot \log \left(\sum_{i=1}^m \sqrt{p_i} \right) + 2p \log(1/p).$$

Proof:

$$\sum_{i=1}^m p_i \cdot \log \frac{1}{p_i} = 2 \cdot \sum_{i=1}^m p_i \cdot \log \frac{1}{\sqrt{p_i}} = 2p \cdot \sum_{i=1}^m \frac{p_i}{p} \cdot \log \frac{1}{\sqrt{p_i}} \leq 2p \cdot \log \left(\sum_{i=1}^m \sqrt{p_i}/p \right)$$

where in the last inequality we applied Jensen's inequality relying on the concavity of the log function and the fact that p_i/p is a probability distribution. \square

Proof of Theorem 6.7. For each $k = 0, \dots, n$, we denote the contribution from sets of size k to $\mathbb{H}[f]$ by

$$\mathbb{H}_k[f] \triangleq \sum_{S:|S|=k} \hat{f}(S)^2 \cdot \log(1/\hat{f}(S)^2).$$

We apply Lemma 6.9 on the sequence of numbers $(\hat{f}(S)^2)_{S:|S|=k}$ to get

$$\begin{aligned} \mathbb{H}_k[f] &= \sum_{S:|S|=k} \hat{f}(S)^2 \cdot \log(1/\hat{f}(S)^2) \\ &\leq 2 \cdot \mathbb{W}^k[f] \cdot \log \left(\sum_{S:|S|=k} |\hat{f}(S)| \right) + 2 \cdot \mathbb{W}^k[f] \cdot \log(1/\mathbb{W}^k[f]). \end{aligned}$$

We invoke the bound from Theorem 5.7, $\sum_{S:|S|=k} |\hat{f}(S)| \leq (Cs)^k$ for some universal constant $C > 0$, to get

$$\mathbb{H}_k[f] \leq 2 \cdot \mathbb{W}^k[f] \cdot \log((Cs)^k) + 2 \cdot \mathbb{W}^k[f] \cdot \log \frac{1}{\mathbb{W}^k[f]}.$$

Summing $\mathbb{H}_k[f]$ over $k = 0, 1, \dots, n$ we get

$$\mathbb{H}[f] \leq \sum_{k=0}^n \mathbb{H}_k[f] \leq 2 \cdot \log(Cs) \cdot \sum_{k=0}^n \mathbb{W}^k[f] \cdot k + 2 \cdot \sum_{k=0}^n \mathbb{W}^k[f] \cdot \log \frac{1}{\mathbb{W}^k[f]}.$$

Using the equality $\sum_k \mathbb{W}^k[f] \cdot k = \mathbb{I}[f]$ and the bound from Theorem 6.8 we get

$$\mathbb{H}[f] \leq 2 \cdot \log(Cs) \cdot \mathbb{I}[f] + 2 \cdot 3 \cdot \mathbb{I}[f] = \mathbb{I}[f] \cdot (2 \log s + O(1)).$$

\square

6.3 The switching lemma for DNFs via moments?

Our last application concerns the class of width- w DNF formulas. In Section 3 we showed how the switching lemma implies sensitivity moment bounds for DNFs (and AC^0). Here we show the converse, how a version of the switching lemma can be derived using sensitivity moment bounds. We start by showing that moment bounds for DNFs can be derived from the Satisfiability Coding Lemma of [PPZ97], who give the following tail bound for the sensitivity:

Lemma 6.10. [PPZ97] *Let $f : \{0, 1\}^n \rightarrow \{\pm 1\}$ be computable by a width- w DNF formula. Then for $s > w$,*

$$\Pr_{\mathbf{x} \leftarrow \{0,1\}^n} [s(f, \mathbf{x}) \geq s] \leq 2^{-s/w}.$$

More precisely, their Satisfiability Coding Lemma [PPZ97, Lemma 2] uses the width- w DNF to construct a randomized encoding where a satisfying input x has (expected) description length bounded by $(n - s(f, x)/w)$, which then implies the above tail bound (see [PPZ97, Fact 3, Lemma 4]). Lemma 6.11 uses this tail bound to derive a moment bound. Lemma 7.3 in Section 7 shows that this bound is tight up to the constant c .

Lemma 6.11. *There exists a constant c such that for every integer $k \geq 1$ and every $f : \{0, 1\}^n \rightarrow \{\pm 1\}$ that is computable by a width- w DNF formula, we have $s^k(f) \leq (ckw)^k$.*

Proof: We have

$$\begin{aligned} s^k(f) &= \sum_{s=1}^n s^k \Pr_{\mathbf{x}} [s(f, \mathbf{x}) = s] \\ &\leq \sum_{\ell=1}^{n/w} (\ell w)^k \Pr_{\mathbf{x}} [s(f, \mathbf{x}) \in \{(\ell-1)w, \dots, \ell w-1\}] \\ &\leq \sum_{\ell=1}^{n/w} (\ell w)^k 2^{-(\ell-1)} \\ &\leq (ckw)^k \end{aligned}$$

for some constant c , where we used Lemma 6.10 to bound $\Pr_{\mathbf{x}} [s(f, \mathbf{x}) \geq (\ell-1)w]$. \square

If Conjecture 4.10 holds, plugging these bounds into Corollary 4.14 gives that there exists $c' > 0$ such that for any width w DNF f ,

$$\Pr_{\rho \leftarrow \mathcal{R}_{k,n}} [\text{dt}(f_{\rho}) \geq k] \leq \frac{8^k s^k(f)}{\binom{n}{k}} \leq \frac{(c'kw)^k}{\binom{n}{k}}.$$

Up to a $k!$ term, this matches the bound from Håstad's switching lemma [Bea94, Lemma 1] which shows

$$\Pr_{\rho \leftarrow \mathcal{R}_{k,n}} [\text{dt}(f_{\rho}) \geq k] \leq \frac{(7kw)^k}{n^k}.$$

Thus proving Conjecture 4.10 would give a combinatorial proof of the switching lemma for DNFs which seems very different from the known proofs of Håstad [Hås86] and Razborov [Raz95].

7 Examples

Lemma 7.1. *Let $n = 2^k - 1$. There exists $f : \{0, 1\}^n \rightarrow \{\pm 1\}$ for which $\text{dt}(f) = \log(n + 1)$ whereas $\text{ts}(f) = n$.*

Proof: Take a complete binary tree with n internal nodes and $n + 1$ leaves. The leaves are alternately labelled 1 and -1 from left to right, while the internal nodes are labelled with x_1, \dots, x_n according to an in-order traversal of the tree. The bound on decision tree depth follows from the definition of f . To lower bound $\text{ts}(f)$, we start at the -1^n input and start flipping bits from -1 to 1 in the order x_1, \dots, x_n . It can be verified that every bit flip changes the value of the function. \square

Lemma 7.2. *There exists a Boolean function g on n variables such that any proper walk for g has length $\Omega(n^2)$.*

Proof: Assume that n is a power of 2 and fix a Hadamard code of length $n/2$. We define an n -variable function g over variables $x_1, \dots, x_{n/2}$ and $y_1, \dots, y_{n/2}$ as follows: if the string $x_1, \dots, x_{n/2}$ equals the i -th codeword in the Hadamard code of length $n/2$, then the output is y_i , otherwise the output is 0. Note that for any $i \neq j$, if n -bit inputs a, b are sensitive to y_i, y_j respectively then the Hamming distance between a and b must be at least $n/4$. Thus any proper walk must flip at least $n/4$ bits between any two vertices that are sensitive to different y_i s, so the minimum length of any proper walk must be at least $n^2/8$. \square

The next example, which shows the tightness of Lemma 6.11 up to the constant c is from [PPZ97] where it is used to show the tightness of Lemma 6.10.

Lemma 7.3. *For every k , there exists a Boolean function h which is computable by a width- w DNF, such that $s^k(h) \geq (kw/2)^k$.*

Proof: Let $x \in \{0, 1\}^{k \times w}$ be a $k \times w$ array of bits. Define the function h to be one if some row contains an odd number of 1s and 0 otherwise. Formally, let

$$h(x) = \bigvee_{i=1}^k \bigoplus_{j=1}^w x_{ij}.$$

To lower bound its moment, note that with probability 2^{-k} , every row contains an even number of 1s. For such x , the $s(h, x) = kw$. This shows that $s^k(h) \geq (kw/2)^k$. \square

On the tightness of the Fourier tail bound

We will construct a low-sensitivity function showing that the Fourier tail bounds in Theorem 1.2 are nearly tight, completing the proof of Lemma 5.9.

For any positive integer r , the Hamming code is a linear code of length $m = 2^r - 1$, with $2^{m-r} = 2^m/(m+1)$ codewords, such that any two codewords are of distance at least 3 from one another. Take HAM_m to be the indicator function of the Hamming code of length m . We have that the maximal sensitivity on 0-inputs for HAM_m (denoted $s_0(\text{HAM}_m)$) equals 1, as any non-codeword may be adjacent to at most one codeword (otherwise the minimal distance of the code would be 2). The maximal sensitivity on 1-inputs for HAM_m (denoted $s_1(\text{HAM}_m)$) equals m , as any codeword is adjacent to m non-codewords in the $\{0, 1\}^m$.

The Hamming code is a linear subspace $U \subseteq \mathbb{F}_2^m$ defined by r linear equations. The dual-code (i.e., the dual subspace U^\perp) is the Hadamard code. There are $m = 2^r - 1$ non-zero vectors in

U^\perp , all of them with Hamming-weight $(m+1)/2$. It is easy to check that the Fourier transform of $\text{HAM}_m : \{0,1\}^m \rightarrow \{0,1\}$ (note that we are viewing HAM_m as a function to $\{0,1\}$ and not $\{-1,1\}$) is

$$\text{HAM}_m(x) = \sum_{\alpha \in U^\perp} \frac{1}{m+1} \cdot (-1)^{\sum_{j=1}^m \alpha_j x_j}.$$

For any integer ℓ , take $f = \text{OR}_m \circ \text{HAM}_m \circ \text{PARITY}_\ell$. To be more precise, the function is defined over $m^2 \cdot \ell$ variables $(x_{i,j,k})_{i \in [m], j \in [m], k \in [\ell]}$ as follows:

$$f(x) = \bigvee_{i=1}^m \text{HAM}_m(y_{i,1}, \dots, y_{i,m}), \quad \text{where} \quad y_{i,j} = \text{PARITY}_\ell(x_{i,j,1}, \dots, x_{i,j,\ell}).$$

Lemma 7.4. *The sensitivity of f is $m \cdot \ell$.*

Proof. The sensitivity of $g = \text{OR}_m \circ \text{HAM}_m$ is at most m since: (1) the sensitivity of 1-inputs of g is at most the sensitivity of 1-inputs of HAM_m , and (2) the sensitivity of 0-inputs of g is at most the sensitivity of 0-inputs of HAM_m times m . To deduce an upper bound on the sensitivity of f we use the fact that for any two functions f_1, f_2 we have $s(f_1 \circ f_2) \leq s(f_1) \cdot s(f_2)$. This gives $s(f) \leq s(g) \cdot s(\text{PARITY}_\ell) \leq m \cdot \ell$.

To show that the sensitivity is at least $m \cdot \ell$ take $x_{i,j,k}$ such that for exactly one i , the vector $(y_{i,1}, \dots, y_{i,m})$ is a codeword of the Hamming-code. The value of f on x is 1, and changing any one of the bits $\{x_{i,j,k} : j \in [m], k \in [\ell]\}$ yields an input x' such that $f(x') = 0$. This shows that the sensitivity of f on x is at least $m \cdot \ell$, completing the proof. \square

Lemma 7.5. *f has Fourier weight at least $0.5 \cdot m^{-m}$ on the $(m \cdot \frac{m+1}{2} \cdot \ell)$ -th layer.*

Proof. We view f as a function $f : \{0,1\}^{m^2\ell} \rightarrow \{0,1\}$ and write it as a multilinear polynomial in its inputs. Since the Fourier transform is the unique multilinear polynomial equaling f on $\{0,1\}^{m^2\ell}$, we can “read” the Fourier coefficients of f from the polynomial. Then we translate this to the Fourier coefficients of $f' : \{0,1\}^{m^2\ell} \rightarrow \{-1,1\}$ defined by $f'(x) = (-1)^{f(x)} = 1 - 2f(x)$.

We write a polynomial agreeing with OR_m on $\{0,1\}^m$:

$$\text{OR}_m(z_1, \dots, z_m) = 1 - \prod_{i=1}^m (1 - z_i) = \sum_{\emptyset \neq S \subseteq [m]} (-1)^{|S|} \prod_{i \in S} z_i.$$

We substitute $\text{HAM}_m(y_{i,1}, \dots, y_{i,m})$ for each z_i and then substitute $(\sum_{k=1}^\ell x_{i,j,k} \bmod 2)$ for each $y_{i,j}$ to get

$$\begin{aligned} f(x) &= \sum_{\emptyset \neq S \subseteq [m]} (-1)^{|S|} \cdot \prod_{i \in S} \text{HAM}_m(y_{i,1}, \dots, y_{i,m}) \\ &= \sum_{\emptyset \neq S \subseteq [m]} (-1)^{|S|} \cdot \prod_{i \in S} \left(\sum_{\alpha_i \in U^\perp} \frac{1}{m+1} \cdot (-1)^{\sum_{j=1}^m \alpha_{i,j} \cdot y_{i,j}} \right) \\ &= \sum_{\emptyset \neq S \subseteq [m]} (-1)^{|S|} \cdot \frac{1}{(m+1)^{|S|}} \cdot \sum_{(\alpha_i)_{i \in S}} (-1)^{\sum_{i \in S} \sum_{j=1}^m \alpha_{i,j} \cdot y_{i,j}} \\ &= \sum_{\emptyset \neq S \subseteq [m]} (-1)^{|S|} \cdot \frac{1}{(m+1)^{|S|}} \cdot \sum_{(\alpha_i)_{i \in S}} (-1)^{\sum_{i \in S} \sum_{j=1}^m \sum_{k=1}^\ell \alpha_{i,j} \cdot x_{i,j,k}}. \end{aligned}$$

Gathering the terms according to the characters gives the Fourier expansion of f . For $S = [m]$ and a fixed set of nonzero vectors $\alpha_1, \dots, \alpha_m$ in U^\perp , the term

$$(-1)^m \cdot \frac{1}{(m+1)^m} \cdot (-1)^{\sum_{i=1}^m \sum_{j=1}^m \sum_{k=1}^\ell \alpha_{i,j} \cdot x_{i,j,k}}$$

appears in the Fourier expansion of f , since it is the only term that appears that is a constant times the character

$$(-1)^{\sum_{i=1}^m \sum_{j=1}^m \sum_{k=1}^\ell \alpha_{i,j} \cdot x_{i,j,k}}.$$

In other words, the Fourier coefficient of f corresponding to the set $\{(i, j, k) : \alpha_{i,j} = 1\}$ equals $(-1)^m / (m+1)^m$. Furthermore, this set is of size $\ell \cdot m \cdot (m+1)/2$, since every α is of weight $(m+1)/2$. It follows that the Fourier-coefficient of $f' = 1 - 2f$ corresponding to the same set equals

$$\frac{-2 \cdot (-1)^m}{(m+1)^m}.$$

As there are m^m choices for non-zero vectors $\alpha_1, \dots, \alpha_m$ in U^\perp , the Fourier weight at level $\ell \cdot m \cdot (m+1)/2$ of f' is at least

$$m^m \cdot \frac{4}{(m+1)^{2m}} \geq 0.5 \cdot m^{-m}$$

for any positive integer m . □

Proof of Lemma 5.9. Choose the maximal integer m such that $0.5 \cdot m^{-m} \geq \epsilon$. This gives $m = \Theta(\frac{\log(1/\epsilon)}{\log \log(1/\epsilon)})$. By assumption on ϵ , $m \leq s$. Choose $\ell = \lfloor s/m \rfloor$ and take $f = \text{OR}_m \circ \text{HAM}_m \circ \text{PARITY}_\ell$ as above. Then, by Lemma 7.4 the sensitivity of f is at most $m \cdot \ell \leq s$. By Lemma 7.5 the weight at level

$$m \cdot \frac{m+1}{2} \cdot \ell = \Theta\left(s \cdot \frac{\log(1/\epsilon)}{\log \log(1/\epsilon)}\right)$$

is at least ϵ . □

8 Open questions

We hope that this work will stimulate further research on the sensitivity graph G_f and on complexity measures associated with it. In this context, we'd like to highlight Conjecture 4.10 which we feel is of independent interest, and if true, implies the robust sensitivity conjecture (Conjecture 1.3) by Corollary 4.15.

Another natural question is whether the reverse direction of the robust sensitivity conjecture also holds: for every k , does there exist a'_k, b'_k such that $\mathbb{E}[s^k] \leq a'_k \mathbb{E}[d^k] + b'_k$? Can one relate this question to a statement about graph-theoretic (or other) complexity measures?

The graph G_f consists of a number of connected components. This component structure naturally suggests another complexity measure:

Definition 8.1. For $x \in \{0,1\}^n$, the component dimension of f at x , denoted $\text{cdim}(f, x)$, is the dimension of the connected component of G_f that contains x (i.e. the number of coordinates i such that x 's component contains at least one edge in the i -th direction). We define $\text{cdim}(f)$ to be $\max_{x \in \{0,1\}^n} \text{cdim}(f, x)$.

It is easy to see that $\text{cdim}(f) \geq \text{ts}(f) \geq s(f)$, and thus a consequence of Conjecture 4.10 is that $\text{cdim}(f) \geq \text{dt}(f)$; however we have not been able to prove a better lower bound for $\text{cdim}(f)$ in terms of $\text{dt}(f)$ than that implied by Theorem 4.9. We note that $\text{cdim}(f)$ and $\text{ts}(f)$ are not polynomially related, since the addressing function shows that the gap between them can be exponential.

The problem of PAC-learning low-sensitivity functions under arbitrary distributions is an intriguing open problem. Conjecture 1.1 implies that it should be possible to PAC learn the class of sensitivity- s functions in time $n^{\text{poly}(s)}$ under arbitrary distributions. While we have shown this holds true under the uniform distribution, we do not know how to do better than $n^{\exp(O(s))}$ for arbitrary distributions.

Acknowledgments

We thank Yuval Peres and Laci Lovasz for pointing us to Sidorenko’s theorem and the related literature. We also thank Yuval for useful discussions, and thank David Levin and Yuval Peres for letting us present the proof of Lemma 4.12 here. We thank Gagan Aggarwal for showing us a combinatorial proof of Sidorenko’s theorem. We thank D. Sivakumar for discussions about Section 6.1, and for drawing our attention to relevant work in machine learning.

References

- [ABG⁺14] A. Ambainis, M. Bavarian, Y. Gao, J. Mao, X. Sun, and S. Zuo. Tighter relations between sensitivity and other complexity measures. In *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014*, pages 101–113, 2014. 1
- [AP14] A. Ambainis and K. Prusis. A tight lower bound on certificate complexity in terms of block sensitivity and sensitivity. In *MFCS*, pages 33–44, 2014. 1
- [APV15] A. Ambainis, K. Prusis, and J. Vihrovs. Sensitivity versus certificate complexity of boolean functions. *CoRR*, abs/1503.07691, 2015. 1
- [AV15] A. Ambainis and J. Vihrovs. Size of Sets with Small Sensitivity: a Generalization of Simon’s Lemma. In *Theory and Applications of Models of Computation - 12th Annual Conference, TAMC 2015*, pages 122–133, 2015. 1
- [BCV13] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1798–1828, 2013. 6.1
- [BdW02] H. Buhrman and R. de Wolf. Complexity measures and decision tree complexity: a survey. *Theoretical Computer Science*, 288(1):21–43, 2002. 1.1
- [Bea94] P. Beame. A switching lemma primer. 1994. 6.3
- [BI87] M. Blum and R. Impagliazzo. Generic oracles and oracle classes. In *Proceedings of the 28th Annual Symposium on Foundations of Computer Science*, pages 118–126, 1987. 4.1
- [BL07] Y. Bengio and Y. LeCun. Scaling learning algorithms towards AI. In Léon Bottou, Olivier Chapelle, D. DeCoste, and J. Weston, editors, *Large Scale Kernel Machines*. MIT Press, 2007. 6.1

- [CKLS15] S. Chakraborty, R. Kulkarni, S. V. Lokam, and N. Saurabh. Upper bounds on fourier entropy. In *Computing and Combinatorics - 21st International Conference, COCOON 2015*, pages 771–782, 2015. 1.1, 2
- [CL14] P. Csikvári and Z. Lin. Graph homomorphisms between trees. *Electronic Journal of Combinatorics*, 21(4):P4.9, 2014. 4.2
- [FK96] E. Friedgut and G. Kalai. Every monotone graph property has a sharp threshold. *Proceedings of the American mathematical Society*, 124(10):2993–3002, 1996. 1.1, 1.3, 6.2
- [Fri98] E. Friedgut. Boolean functions with low average sensitivity depend on few coordinates. *Combinatorica*, 18(1):474–483, 1998. 1.1
- [GNS⁺16] P. Gopalan, N. Nisan, R. Servedio, K. Talwar, and A. Wigderson. Smooth boolean functions are easy: efficient algorithms for low sensitivity functions. In *ITCS*, pages 59–70, 2016. 1, 6.1, 6.1, 6.2, 6.1, 6.4
- [GSW16a] P. Gopalan, R. Servedio, and A. Wigderson. Degree and Sensitivity: tails of two distributions. In *Conference on Computational Complexity (CCC’2016): to appear*, 2016. *, 1
- [GSW16b] P. Gopalan, R. Servedio, and A. Wigderson. Manuscript in preparation, 2016. 6.1
- [Hås86] J. Håstad. *Computational Limitations for Small Depth Circuits*. MIT Press, Cambridge, MA, 1986. 3, 6.3
- [HH91] J. Hartmanis and L.A. Hemachandra. One-way functions, robustness and non-isomorphism of NP-complete classes. *Theor. Comput. Sci*, 81(1):155–163, 1991. 4.1
- [HKP11] P. Hatami, R. Kulkarni, and D. Pankratov. *Variations on the Sensitivity Conjecture*. Number 4 in Graduate Surveys. Theory of Computing Library, 2011. 1.1
- [KK04] C. Kenyon and S. Kutin. Sensitivity, block sensitivity, and l-block sensitivity of Boolean functions. *Information and Computation*, pages 43–53, 2004. 1
- [KKL88] J. Kahn, G. Kalai, and N. Linial. The influence of variables on boolean functions. In *Proc. 29th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 68–80, 1988. 1.1
- [KOS04] A. Klivans, R. O’Donnell, and R. Servedio. Learning intersections and thresholds of halfspaces. *Journal of Computer & System Sciences*, 68(4):808–840, 2004. 6.1, 6.1
- [LMN93] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, Fourier transform and learnability. *Journal of the ACM*, 40(3):607–620, 1993. 1.1, 1.3, 3, 3, 6.1, 6.1
- [LP16] D. Levin and Y. Peres. Counting walks and graph homomorphisms via Markov chains and importance sampling. Manuscript, 2016. 4.2, 4.12
- [Nis91] N. Nisan. CREW PRAMs and decision trees. *SIAM Journal on Computing*, 20(6):999–1007, 1991. 1, 1.1, 1.1

- [NS94] N. Nisan and M. Szegedy. On the degree of Boolean functions as real polynomials. *Comput. Complexity*, 4:301–313, 1994. 1, 1.1
- [OWZ11] R. O’Donnell, J. Wright, and Y. Zhou. The fourier entropy-influence conjecture for certain classes of boolean functions. In *Automata, Languages and Programming - 38th International Colloquium , ICALP 2011*, pages 330–341, 2011. 6.2, 6.8
- [PPZ97] R. Paturi, P. Pudlák, and F. Zane. Satisfiability coding lemma. In *38th Annual Symposium on Foundations of Computer Science, FOCS ’97*, pages 566–574, 1997. 3, 6.3, 6.10, 6.3, 7
- [Raz95] A. A. Razborov. Bounded arithmetic and lower bounds in Boolean complexity. In *Feasible Mathematics II*, pages 344–386. Springer, 1995. 5.2, 6.3
- [Sid94] A. Sidorenko. A partially ordered set of functionals corresponding to graphs. *Discrete Mathematics*, 131(1-3):263–277, 1994. 1.2, 4.2
- [Sim82] H. U. Simon. A tight $\Omega(\log \log n)$ -bound on the time for parallel ram’s to compute nondegenerated boolean functions. *Information and Control*, 55(1-3):102–106, 1982. 6.1
- [Tal14a] A. Tal. Shrinkage of de Morgan formulae from quantum query complexity. In *FOCS*, pages 551–560, 2014. 1.2, 3
- [Tal14b] A. Tal. Tight Bounds on The Fourier Spectrum of AC^0 . ECCC report TR14-174 Revision #1, available at <http://eccc.hpi-web.de/report/2014/174/>, 2014. 1.1, 3, 5.3
- [Tar89] G. Tardos. Query complexity, or why is it difficult to separate $NP^A \cap co - NP^A$ from P^A by a random oracle A ? *Combinatorica*, 8(4):385–392, 1989. 4.1